Submitted to
manuscript (Please, provide the manuscript number!)

Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs

Jing Xie
School of Operations Research & Information Engineering, Cornell University, Ithaca, NY 14853, jx66@cornell.edu

Peter I. Frazier
School of Operations Research & Information Engineering, Cornell University, Ithaca, NY 14853, pf98@cornell.edu

Stephen E. Chick
Technology & Operations Management Area, INSEAD, Boulevard de Constance, 77300 Fontainebleau, FRANCE, stephen.chick@insead.edu

This paper addresses discrete optimization via simulation. We show that allowing for both a correlated prior distribution on the means (e.g., with discrete kriging models) and sampling correlation (e.g., with common random numbers, or CRN) can significantly improve the ability to identify the best alternative. These two correlations are brought together for the first time in a highly-sequential knowledge-gradient sampling algorithm, which chooses points to sample using a Bayesian value of information (VOI) criterion. We provide almost sure convergence guarantees as the number of samples grows without bound when parameters are known, provide approximations that allow practical implementation, and demonstrate that CRN leads to improved optimization performance for VOI-based algorithms in sequential sampling environments with a combinatorial number of alternatives and costly samples.

Key words: discrete optimization via simulation; value of information; kriging model

---

We consider discrete optimization via simulation, in which we have a discrete set of alternative systems whose performance can each be evaluated via stochastic simulation, and we wish to allocate a limited simulation budget among them to find one whose expected performance is as large as possible. Because of its importance, previous authors have proposed algorithms of several types to address this problem, including randomized search (Andradóttir 1998, 2006, Zhou et al. 2008), metaheuristics (Shi and Ólafsson 2000), metamodel-based algorithms (Barton 2009, van Beers and Kleijnen 2008), Bayesian value-of-information algorithms (Chick 2006, Frazier 2010), local search algorithms (Wang et al. 2013, Hong and Nelson 2006, Xu et al. 2010), model-based search (Hu et al. 2012, Wang et al. 2010), and ranking and selection algorithms (Kim and Nelson 2006, Chen and Lee 2010, Branke et al. 2007). Andradóttir (1998) and Fu (2002) provide surveys of the field.

We study this problem in a Bayesian context, where we place a prior probability distribution on the values of the alternatives, and use value of information (VOI) calculations within a knowledge-gradient (KG) sampling algorithm to decide which alternative, or collection of alternatives, would be most useful to sample next. The advantage of doing so is that making decisions based on the VOI automatically addresses the exploration versus exploitation tradeoff, and tends to reduce the number of function evaluations required on average to reach a given solution quality, potentially (but not necessarily) at the cost of requiring more computation to decide where to sample.

The prior probability distribution that we consider is a multivariate normal distribution, and allows for correlation in our prior belief between two alternatives. This models a belief that two alternatives with similar characteristics often have similar expected performance, and allows the algorithm that we construct to do well even in problems where the number of alternatives is much larger than the number of samples that we can take.

We allow common random numbers (CRN), in which multiple alternatives are simulated using the same stream of random numbers. This induces correlation in the noise, which can be advantageous for optimization when the correlation is positive, because it allows more accurate estimation of the differences between alternatives' values.

Several previous authors have considered Bayesian formulations of optimization via simulation. The setting most frequently studied is that of ranking and selection, with relatively few alternatives, an independent prior distribution, and independent sampling (Gupta and Miescke 1996, Chick and Inoue 2001b, Frazier et al. 2008, Chick and Frazier 2012). Bayesian optimization via simulation with correlated prior distributions (but not with CRN) for problems with many alternatives was considered in a discrete setting (Frazier et al. 2009) and in a continuous setting (Villemonteix et al. 2009, Huang et al. 2006, Scott et al. 2011). This work in a continuous setting parallels work on noise-free Bayesian global optimization (Jones et al. 1998, Forrester et al. 2008, Brochu et al. 2009).

Our analysis differs from this previous literature by allowing the use of CRN. This has been perceived to be difficult, because sampling with CRN makes it difficult to compute the VOI, and to maintain a closed-form posterior distribution. We overcome these difficulties by calculating the VOI for observing the *difference* in value between two alternatives, which can be done analytically, and by calculating the posterior with adaptively updated point estimates of the noise covariance. We show that, in the context of VOI-based algorithms, using CRN can greatly improve performance.

Sampling with correlated means and CRN in the Bayesian setting using VOI methods has been considered by Chick and Inoue (2001a), but assumed two-stage sampling rather than fully sequential sampling, and restricted attention to conjugate prior distributions for the unknown means. Others have considered sampling with CRN in the optimal computing budget allocation framework (Fu et al. 2004), in the indifference-zone setting (Clark and Yang 1986, Nelson and

Matejcik 1995), and in the multiple comparisons problem (Yang and Nelson 1991, Nakayama 2000, Kim 2005). The current work differs from this previous work in its focus on problems with many alternatives, enabled by a multivariate normal prior distribution with arbitrary covariance.

The current work, in its use of multivariate normal prior distributions, makes a link to Gaussian process (GP) priors (Rasmussen and Williams 2006) and stochastic kriging (Ankenman et al. 2010, Chen et al. 2012, 2013). When alternatives correspond to points on a grid, as they do in many resource allocation problems (e.g., each alternative specifies the number of each of several employee types to have present), our use of a multivariate normal prior distribution can be implemented by placing a GP prior over the continuum, and then only considering points on the grid.

We present three techniques that reduce the computation required to find a point, or pair of points, with a large VOI. The first is to use the gradient of the VOI in performing this search, calculating it over an embedding of our discrete alternatives into a continuous space. This use of the gradient of the VOI differs from the more common use of gradients of the response surface in optimization. The second is to consider a VOI with a restricted set of implementation decisions. The third is to use data structures that avoid enumerating alternatives, instead tracking only those alternatives that have been sampled, and reconstructing required portions of the posterior distribution as needed. This is standard in GP regression, but contrasts with previous work on optimization via simulation with CRN (Clark and Yang 1986, Nelson and Matejcik 1995, Chick and Inoue 2001a, Fu et al. 2004). These three techniques were applied in Scott et al. (2011) to a continuous setting without CRN.

We also provide an almost sure guarantee of convergence to the global optimum, as the number of samples taken grows without bound, when parameters are known. In addition to allowing correlated sampling, this theoretical result contrasts with Scott et al. (2011) in having conditions that are easier to verify. It also contrasts with other work that focuses on convergence to local optima (Hong and Nelson 2006, Xu et al. 2010, Wang et al. 2013).

The current paper extends a report of our preliminary work (Frazier et al. 2011) in a number of ways. It provides an enhanced version of the algorithm that scales to much larger problems, a theoretical analysis showing convergence to a global optimum, a derivation of a maximum likelihood estimation method for estimating covariance parameters from samples observed with CRN, and additional numerical comparisons with other algorithms on larger problems.

We begin in §1 by formally defining our problem and the statistical model in which we perform inference. §2 describes a generic sampling algorithm that forms the basis for specific sampling algorithms defined later in the paper. §3 defines the VOI and the corresponding KG factor, and shows how it can be computed in the context of optimization via simulation with correlated sampling. §4 takes these VOI and KG computations, and uses them to create allocation rules for the

KG sampling algorithm. §5 states theoretical results on consistency of KG algorithms, showing that these algorithms can produce consistent estimates of the global optimum in the limit as the sampling budget grows large, when parameters are known. §6 discusses practical implementation issues, regarding prior distributions and computation of the KG algorithm's decisions. Numerical results in §7 show a distinct advantage to the ability to sequentially sample with CRN in discrete optimization via simulation problems. Appendices prove theoretical results and derive gradient and statistical estimation results used in the algorithm.

1. Sampling Model and Mechanism for Posterior Inference

Consider a collection of k alternatives with stochastic performance. If we sample from all k alternatives together using CRN, then we observe a normal random vector. Let the mean vector of this normal distribution be $\theta = [\theta(1), \ldots, \theta(k)]^T$, and let its covariance matrix be Λ, where ^{T} denotes matrix transposition. We wish to find the alternative x with the largest sampling mean $\theta(x)$.

We use a Bayesian formulation, in which we begin with a multivariate normal prior on θ,

$$\theta \sim \mathcal{N}(\mu_0, \Sigma_0). \tag{1}$$

The choice of Σ_0 allows for conjugate prior distributions for θ (Chick and Inoue 2001a) or for GP priors (Rasmussen and Williams 2006), which are related to kriging models (Cressie 1993). A parametric family can be used to specify μ_0 and Σ_0 in terms of a function taking the alternatives and few additional parameters as arguments. In practice, the parameters specifying μ_0 and Σ_0, as well as the sampling covariance Λ, are unknown, but we will initially assume they are fully known for simplicity. Then, we will relax this assumption in §6.

In this paper, the ith entry of a length-k vector v (e.g., θ and μ_0) is written $v(i)$, and the (i,j)th entry of a k-by-k matrix M (e.g., Σ_0 and Λ) is written $M(i,j)$. Moreover, for an ordered collection of m alternatives $\vec{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$ with elements $x^{(i)} \in \{1, 2, \ldots, k\}$ for each i, we use $v(\vec{x})$ to denote the length-m sub-vector of v with the ith entry equal to $v\left(x^{(i)}\right)$. Let $\vec{x}'$ with elements in $\{1, 2, \ldots, k\}$ be another vector of alternatives with m' entries. We denote by $M(\vec{x}, \vec{x}')$ the m-by-m' sub-matrix of M with the (i,j)th entry equal to $M\left(x^{(i)}, x'^{(j)}\right)$.

1.1. Sampling Model and Distribution of Outputs

At each time $n = 1, 2, \ldots$ we choose a set of the alternatives to sample, specified as a row vector $\vec{x}_n$ with elements in $\{1, 2, \ldots, k\}$, and sample each of the chosen alternatives once using CRN. Each alternative may appear at most once in $\vec{x}_n$. We then observe a column vector $\vec{y}_n$, with one entry for each alternative sampled. The conditional distribution of $\vec{y}_n$ given $\vec{x}_n$, θ is assumed to be Gaussian and independent of previous observations,

$$\vec{y}_n \mid \theta, \vec{x}_n, (\vec{x}_m, \vec{y}_m : m < n) \sim \mathcal{N}\left(\theta\left(\vec{x}_n\right), \Lambda\left(\vec{x}_n, \vec{x}_n\right)\right). \tag{2}$$

Although (2) is general, in our algorithm below, the sampling decision $\vec{x}_n$ is either a singleton x_n, with corresponding observation y_n, or a pair of alternatives $\left(x_n^{(1)}, x_n^{(2)}\right)$, with corresponding observations $(y_n^{(1)}, y_n^{(2)})$. The notation $\vec{x}_n$ and $\vec{y}_n$ indicates the general case, in which one or more alternatives is sampled, while x_n and y_n always indicates a single alternative. The sampling distribution of (2) for these two cases (singletons and pairs) are

$$\begin{aligned} y_n \mid \theta, x_n &\sim \mathcal{N}\left(\theta(x_n), \Lambda(x_n, x_n)\right), \quad \text{and} \\ (y_n^{(1)}, y_n^{(2)}) \mid \theta, \left(x_n^{(1)}, x_n^{(2)}\right) &\sim \mathcal{N}\left(\begin{bmatrix}\theta\left(x_n^{(1)}\right)\\ \theta\left(x_n^{(2)}\right)\end{bmatrix}, \begin{bmatrix}\Lambda\left(x_n^{(1)}, x_n^{(1)}\right) & \Lambda\left(x_n^{(1)}, x_n^{(2)}\right)\\ \Lambda\left(x_n^{(2)}, x_n^{(1)}\right) & \Lambda\left(x_n^{(2)}, x_n^{(2)}\right)\end{bmatrix}\right). \end{aligned}$$

These sampling distributions are sufficient for calculating posterior distributions from observations in the sampling algorithms that we propose, but when computing the VOI in §3 below, we will also consider three additional sampling distributions. First, we will consider the sampling distribution of observing only the *difference* between a pair $\left(x_n^{(1)}, x_n^{(2)}\right)$ of alternatives,

$$y_n^{(1)} - y_n^{(2)} \mid \theta, \left(x_n^{(1)}, x_n^{(2)}\right) \sim \mathcal{N}\left(\theta\left(x_n^{(1)}\right) - \theta\left(x_n^{(2)}\right), \Lambda\left(x_n^{(1)}, x_n^{(1)}\right) + \Lambda\left(x_n^{(2)}, x_n^{(2)}\right) - 2\Lambda\left(x_n^{(1)}, x_n^{(2)}\right)\right).$$

Second, we will consider the sampling distribution of observing not necessarily one but $\beta_n \geq 1$ vectors of samples from the distribution given by (2), each generated using an independent CRN stream. We do this to compute an average VOI per sample. The value of β_n can be fixed beforehand, or can be chosen adaptively. We generalize $\vec{y}_n$ to refer to the average of these β_n observations, so

$$\vec{y}_n \mid \theta, \vec{x}_n, \beta_n \sim \mathcal{N}\left(\theta\left(\vec{x}_n\right), \Lambda\left(\vec{x}_n, \vec{x}_n\right)/\beta_n\right). \tag{3}$$

Third, we will consider the sampling distribution of observing $\beta_n \geq 1$ independent differences between a pair $\left(x_n^{(1)}, x_n^{(2)}\right)$, continuing to let $\vec{y}_n = (y_n^{(1)}, y_n^{(2)})$ denote the average of these observations,

$$y_n^{(1)} - y_n^{(2)} \mid \theta, \left(x_n^{(1)}, x_n^{(2)}\right), \beta_n \sim \mathcal{N}\left(\theta\left(x_n^{(1)}\right) - \theta\left(x_n^{(2)}\right), \left[\Lambda\left(x_n^{(1)}, x_n^{(1)}\right) + \Lambda\left(x_n^{(2)}, x_n^{(2)}\right) - 2\Lambda\left(x_n^{(1)}, x_n^{(2)}\right)\right]/\beta_n\right). \tag{4}$$

These last three sampling distributions are used only to compute the VOI. In the sampling algorithms that we propose, we always observe from both alternatives when sampling from a pair, and take only one sample at a time from a singleton or pair even when we calculate a VOI with $\beta_n > 1$.

1.2. Posterior Distribution for Unknown Means and its Computation

With the sampling scheme in (2), and the assumption that the sampling covariance matrix Λ is known, we can compute a closed-form expression for the posterior distribution on θ. We let $\mathbb{E}_n$ and Var_n indicate the conditional expectation and variance respectively with respect to the data $\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2, \ldots, \vec{x}_n, \vec{y}_n$, where each $\vec{y}_n$ is sampled according to (2). Define $\mu_n = \mathbb{E}_n\theta$ and $\Sigma_n = \mathrm{Var}_n\theta$. The posterior distribution on θ is normal (see, e.g., Gelman et al. 2004, Sec. 14.6),

$$\theta \mid \vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2, \ldots, \vec{x}_n, \vec{y}_n \sim \mathcal{N}\left(\mu_n, \Sigma_n\right),$$

where the posterior mean μ_n and variance Σ_n can be computed analytically, either directly from the prior and the full data, or recursively, updating as each new datapoint $\vec{x}_n, \vec{y}_n$ is added.

When the number of alternatives k is large, it is computationally infeasible to store all of μ_n and Σ_n, because Σ_n is a k-by-k matrix. Therefore, we use a method commonly used in GP regression, which calculates the posterior distribution on the sampled alternatives and any desired additional alternatives, without requiring a k-by-k matrix. We briefly describe this method here, giving some notation to be used later, and focusing on singletons and pairs.

Let $\mathcal{X}_n$ denote the cumulative row vector of alternatives sampled from time 1 to time n, i.e., the concatenation of $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ into a row. Alternatives appear more than once if they are sampled more than once. For example, if $\vec{x}_1 = x_1$ and $\vec{x}_2 = \left(x_2^{(1)}, x_2^{(2)}\right)$, then $\mathcal{X}_1 = (x_1)$ and $\mathcal{X}_2 = \left(x_1, x_2^{(1)}, x_2^{(2)}\right)$. In addition, if $x_1 = x_2^{(1)} = x$ then $\mathcal{X}_2 = \left(x, x, x_2^{(2)}\right)$.

In §3 we will compute the VOI for an arbitrary (singleton or pair) sampling decision $\vec{x}$ at time $n+1$. Let the vector $\mathcal{X}_{n,\vec{x}}$ denote the row concatenation of $\mathcal{X}_n$ and $\vec{x}$. To compute the VOI, we require the posterior distribution on $\theta\left(\mathcal{X}_{n,\vec{x}}\right)$, which is multivariate normal with mean $\mu_n\left(\mathcal{X}_{n,\vec{x}}\right)$ and covariance $\Sigma_n\left(\mathcal{X}_{n,\vec{x}}, \mathcal{X}_{n,\vec{x}}\right)$. We introduce the following expressions for computing these quantities. Let $\mathcal{Y}_n$ be the cumulative column vector of sampling observations up to time n, i.e., the columnar concatenation of $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_n$, so each entry of $\mathcal{Y}_n$ is the observation from the corresponding entry in $\mathcal{X}_n$. Let Γ_n be the block diagonal matrix with n blocks: $\Lambda(\vec{x}_1, \vec{x}_1), \Lambda(\vec{x}_2, \vec{x}_2), \ldots, \Lambda(\vec{x}_n, \vec{x}_n)$. We then define three quantities, the measurement residual $\widetilde{\mathcal{Y}}_n$, the residual covariance S_n, and the optimal Kalman gain $K_n(\vec{x})$, by

$$\widetilde{\mathcal{Y}}_n = \mathcal{Y}_n - \mu_0\left(\mathcal{X}_n\right), \qquad S_n = \Sigma_0\left(\mathcal{X}_n, \mathcal{X}_n\right) + \Gamma_n, \qquad K_n(\vec{x}) = \Sigma_0\left(\mathcal{X}_{n,\vec{x}}, \mathcal{X}_{n,\vec{x}}\right) L\left[S_n\right]^{-1}. \tag{5}$$

Here, the matrix L is defined by concatenating an $|\mathcal{X}_n|$-by-$|\mathcal{X}_n|$ identify matrix with an $|\mathcal{X}_n|$-by-$|\vec{x}_n|$ matrix of zeros, so $L = [I_{|\mathcal{X}_n|}, \vec{0}]^T$ if $\vec{x} = x$, and $L = [I_{|\mathcal{X}_n|}, \vec{0}, \vec{0}]^T$ if $\vec{x} = \left(x^{(1)}, x^{(2)}\right)$. Here and elsewhere, $|\cdot|$ denotes the length of a vector. We will assume in §5 that Σ_0 and Λ are positive definite. That assumption implies that $\Sigma_0\left(\mathcal{X}_n, \mathcal{X}_n\right)$ is positive semidefinite and that Γ_n is positive definite, so that S_n is positive definite and that its inverse $[S_n]^{-1}$ exists.

The posterior mean and covariance matrix of $\theta\left(\mathcal{X}_{n,\vec{x}}\right)$ at time n are then given respectively by

$$\mu_n\left(\mathcal{X}_{n,\vec{x}}\right) = \mu_0\left(\mathcal{X}_{n,\vec{x}}\right) + K_n(\vec{x})\widetilde{\mathcal{Y}}_n, \tag{6}$$

$$\Sigma_n\left(\mathcal{X}_{n,\vec{x}}, \mathcal{X}_{n,\vec{x}}\right) = \left(I_{|\mathcal{X}_{n,\vec{x}|}} - K_n(\vec{x})L^T\right)\Sigma_0\left(\mathcal{X}_{n,\vec{x}}, \mathcal{X}_{n,\vec{x}}\right). \tag{7}$$

In implementing (5), one should not invert S_n directly, as doing so when n is large is numerically unstable. Instead, one can perform a Cholesky decomposition, and then solve a numerical system, as is described in Sec. 2.2 of Rasmussen and Williams (2006). This is more stable, and faster. For further discussion of implementation issues in GP regression, see Rasmussen and Williams (2006).

2. Generic Sampling Algorithm

We now formalize our proposed DOvS algorithm. The notation in §1 allows us to formalize it in a way that is amenable to handling a very large number of alternatives: statistics are tracked only for alternatives that have been sampled or are being considered for sampling in the next stage.

The algorithm samples in a sequential manner. This requires the specification of an allocation rule, which maps $\mathcal{X}_n, \mathcal{Y}_n$ to a set of alternatives to sample next, and a stopping rule, which decides whether or not to stop sampling. The allocation rules we use are based on VOI principles described in §3 and are presented in §4. The default stopping rule we use in this paper is to stop after a pre-specified number of samples is observed.

The generic algorithm below is written to be able to handle either a known or an unknown sampling covariance matrix Λ. When it is unknown, as is typical in applications, the sampling covariance parameters are estimated. In this case, we also maintain estimates of the parameters μ_0 and Σ_0 defining the prior distribution in an empirical Bayes fashion, as described below.

1. **Initialize:** Select an allocation rule and a stopping rule. If the sampling covariance Λ and the mean vector μ_0 and the covariance matrix Σ_0 for the unknown sampling means θ are known, then specify these parameters, initialize $n = 0$ to be the number of stages of sampling done so far, and initialize $\mathcal{X}_0$ and $\mathcal{Y}_0$ to be empty vectors. If Λ, μ_0 and Σ_0 are not all known, then describe the functional forms of Λ, μ_0 and Σ_0 in terms of a collection of parameters (see §6.1), and take an initial stage of samples to estimate those parameters, setting n, $\mathcal{X}_n$ and $\mathcal{Y}_n$ accordingly (see §6.2).

2. **Update parameters (Empirical Bayes):** If the parameters determining Λ are unknown and their estimates are to be updated, then use the maximum likelihood estimator described in §6.2 to estimate them using all data (collected in $\mathcal{X}_n$ and $\mathcal{Y}_n$).

3. **Check allocation and stopping rule:** If the stopping rule says to stop sampling, go to Step 5. Otherwise, use the allocation rule to choose a set of alternatives, $\vec{x}_{n+1}$, to sample next.

4. **Sample:** Sample $\vec{y}_{n+1}$ using CRN according to (2) with the chosen $\vec{x}_{n+1}$. Concatenate $\vec{y}_{n+1}$ with $\mathcal{Y}_n$ to get $\mathcal{Y}_{n+1}$, and $\vec{x}_{n+1}$ with $\mathcal{X}_n$ to get $\mathcal{X}_{n+1}$. Increment n and go back to Step 2.

5. **Selection rule:** Select as the best the alternative in $\mathcal{X}_n$ with the largest posterior mean. The can be found by computing $\mu_n(\mathcal{X}_n)$ according to (6) with $\mathcal{X}_{n,\vec{x}} = \mathcal{X}_n$, and then taking the largest component of this vector.

3. Value of Information

In this section we derive analytic expressions for computing the VOI, resulting from sampling singletons, or sampling the difference between pairs of alternatives. These VOI calculations are then used to derive our allocation rules in §4 for use in the algorithm of §2.

VOI is a concept which encompasses the expected value of sample information (EVSI) and the expected value of perfect information (EVPI) (Raiffa and Schlaifer 1961). Information is valued

according to the expected improvement it produces in some decision to be made later. In this paper, the decision to be made later is which alternative to select as the best and to implement in reality. We call this decision the "implementation decision." The value of an implementation decision x is $\theta(x)$ and has expectation $\mu_{n+1}(x)$ under the posterior at time $n+1$. Thus, the expected value of the best implementation decision that can be made at time $n+1$ is $\max_{x\in\{1,2,\ldots,k\}} \mu_{n+1}(x) = \max \mu_{n+1}$. The increment in this value in going from time n to time $n+1$ is $\max \mu_{n+1} - \max \mu_n$ and depends on y_{n+1}. Here, the VOI is the expected value of this increment, under the posterior at time n, under the hypothetical that an alternative is to be selected after a single stage of sampling.

In this framework, the VOI for a set of β samples collected by observing $\vec{y}_{n+1}$ with a general sampling decision $\vec{x}$ at time $n+1$ according to (2) can be written

$$V_n(\vec{x},\beta) = \mathbb{E}_n\left[\max \mu_{n+1} \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta\right] - \max \mu_n. \tag{8}$$

If the implementation decision is restricted to a set $A_n(\vec{x})$ that may depend upon on $\mathcal{X}_n, \mathcal{Y}_n$ and $\vec{x}$, then the VOI is

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = \mathbb{E}_n\left[\max\left[\mu_{n+1}\left(A_n(\vec{x})\right)\right] \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta\right] - \max\left[\mu_n\left(A_n(\vec{x})\right)\right]. \tag{9}$$

When $A_n(\vec{x}) = \{1,2,\ldots,k\}$, then $V_n(\vec{x}, A_n(\vec{x}), \beta) = V_n(\vec{x},\beta)$. This VOI also satisfies a monotonicity property: if $A \subseteq B$ then $V_n(\vec{x}, A, \beta) \leq V_n(\vec{x}, B, \beta)$. This monotonicity property implies that $V_n(\vec{x}, A_n(\vec{x}), \beta)$ is actually a lower bound on $V_n(\vec{x},\beta)$.

There is no restriction on the implementation decision in practice, but we use $V_n(\vec{x}, A_n(\vec{x}), \beta)$ as an approximation to $V_n(\vec{x},\beta)$ because it can be computed more quickly, especially when $|A_n(\vec{x})|$ is small. Methods for choosing $A_n(\vec{x})$ are discussed in §6.3.

3.1. Predictive Distribution for Posterior Means to be Observed

The VOI in (8) or (9) depends on the predictive distribution for $\mu_{n+1}(A)$ that results from a particular decision to sample $\vec{x}_{n+1}$ for β_{n+1} times, for any given set A. We consider two specific types of sampling decisions $\vec{x}_{n+1}$: observing singletons $\vec{x}_{n+1} = (x_{n+1})$ as in (3); and observing the difference between a pair of alternatives $\vec{x}_{n+1} = \left(x^{(1)}_{n+1}, x^{(2)}_{n+1}\right)$ as in (4). Observing either the singleton y_n or the difference $y^{(1)}_n - y^{(2)}_n$ admits an analytic expression for $V_n(\vec{x}, A, \beta)$ below. Observing both $y^{(1)}_n$ and $y^{(2)}_n$ together does not: we use the VOI of sampling their difference as a lower bound on the VOI of observing both values. This lower bound proves to be useful in numerical experiments.

For both singletons and differences between pairs, the predictive distribution is

$$\mu_{n+1}(A) \mid \mathcal{X}_n, \mathcal{Y}_n, \vec{x}_{n+1}, \beta_{n+1} \;\sim\; \mathcal{N}\left(\mu_n(A), \widetilde{\sigma}_n\left(\vec{x}_{n+1}, A, \beta_{n+1}\right) \widetilde{\sigma}_n\left(\vec{x}_{n+1}, A, \beta_{n+1}\right)^T\right), \tag{10}$$

where $\widetilde{\sigma}_n\left(\vec{x}_{n+1}, A, \beta_{n+1}\right)$ is a $|A| \times 1$ vector defined respectively in the two cases as

$$\begin{aligned}\widetilde{\sigma}_n(x, A, \beta) &= \frac{\Sigma_n(A, x)}{\sqrt{\beta^{-1}\Lambda(x, x) + \Sigma_n(x, x)}}, \\ \widetilde{\sigma}_n\left(\left(x^{(1)}, x^{(2)}\right), A, \beta\right) &= \frac{\Sigma_n\left(A, x^{(1)}\right) - \Sigma_n\left(A, x^{(2)}\right)}{\sqrt{\beta^{-1}P + Q_n}},\end{aligned} \tag{11}$$

which follows directly from Frazier et al. (2011, Sec. 2.2). Here, $\Sigma_n(A, x)$ is a column vector containing the entries from Σ_n in column x with rows in A, and P and Q_n are defined by

$$\begin{aligned}P &= \Lambda\left(x^{(1)}, x^{(1)}\right) + \Lambda\left(x^{(2)}, x^{(2)}\right) - 2\Lambda\left(x^{(1)}, x^{(2)}\right), \\ Q_n &= \Sigma_n\left(x^{(1)}, x^{(1)}\right) + \Sigma_n\left(x^{(2)}, x^{(2)}\right) - 2\Sigma_n\left(x^{(1)}, x^{(2)}\right).\end{aligned} \tag{12}$$

This expression will be used in §3.2 to compute the VOI in (9) explicitly.

3.2. Evaluation of the Value of Information

We now provide explicit expressions for the VOI in (9) under observations of singletons and of differences between pairs. From (10), we know that when $\mathcal{X}_n$, $\mathcal{Y}_n$, $\vec{x}_{n+1}$ and β_{n+1} are given, $\mu_{n+1}(A)$ is equal in distribution to $\mu_n(A) + \widetilde{\sigma}_n(\vec{x}_{n+1}, A, \beta_{n+1})Z$, where Z is a standard normal random variable. Using this observation in (9) shows that

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = \mathbb{E}_n\left[\max\left[\mu_n\left(A_n(\vec{x})\right) + \widetilde{\sigma}_n\left(\vec{x}, A_n(\vec{x}), \beta\right) Z\right]\right] - \max\left[\mu_n\left(A_n(\vec{x})\right)\right]. \tag{13}$$

To compute (13), we consider three cases: when $A_n(\vec{x})$ has one, two, or more than two elements. This third case is the most common in the allocation rules developed in §4.

When $A_n(\vec{x})$ has exactly one element, one can show using the tower property of conditional expectation that $V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = 0$. In other words, if only one alternative can ever be selected, information has no value.

When $A_n(\vec{x})$ has exactly two elements, computation of $V_n\left(\vec{x}, A_n(\vec{x}), \beta\right)$ is similar to related computations for the VOI in a pairwise comparison (Frazier et al. 2008, Jones et al. 1998, Chick and Inoue 2001a). Namely, let Δ be the absolute value of the difference of $\mu_n(x)$ between the two different $x \in A_n(\vec{x})$, and let s be the absolute value of the difference of the two components of $\widetilde{\sigma}_n\left(\vec{x}, A_n(\vec{x}), \beta\right)$. Then

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = s f(-\Delta/s),$$

where $f(-z) = \varphi(z) - z\Phi(-z)$, and φ and Φ are the density and cumulative distribution functions, respectively, of a standard normal random variable.

When $A_n(\vec{x})$ contains more than two elements, computation of $V_n\left(\vec{x}, A_n(\vec{x}), \beta\right)$ is more involved, but still can be performed analytically. Recalling (13), we see that we can write

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = h\left(\mu_n\left(A_n(\vec{x})\right), \widetilde{\sigma}_n\left(\vec{x}, A_n(\vec{x}), \beta\right)\right), \tag{14}$$

where $h(a,b) = \mathbb{E}\left[\max_i a(i) + b(i)Z\right] - \max_i a(i)$ for two vectors a and b of equal length. Frazier et al. (2008) gives an exact algorithm for computing h and Frazier (2009–2010) provides a Matlab implementation. More details are given below in §6.3.

In situations where some entries in the sampling covariance Λ are negative, independent sampling for the pairs of alternatives corresponding to these entries is preferred over correlated sampling. More generally, the VOI increases as the sampling correlation increases. This is shown by the following lemma, and is used in our sampling algorithm to improve performance.

LEMMA 1. *Suppose* $\vec{x} = \left(x^{(1)}, x^{(2)}\right)$. *Let* μ_n, Σ_n, $\Lambda\left(x^{(1)}, x^{(1)}\right)$, $\Lambda\left(x^{(2)}, x^{(2)}\right)$ *be fixed. Then for any* A *and* β, $V_n\left(\vec{x}, A, \beta\right)$ *is an increasing function of the sampling correlation between* $x^{(1)}$ *and* $x^{(2)}$,

$$\rho\left(x^{(1)}, x^{(2)}\right) = \frac{\Lambda\left(x^{(1)}, x^{(2)}\right)}{\Lambda\left(x^{(1)}, x^{(1)}\right)\Lambda\left(x^{(2)}, x^{(2)}\right)}.$$

3.3. Knowledge Gradient Factors

The knowledge-gradient (KG) factor is a metric that measures the VOI per sample, when a given alternative $\vec{x}$ is sampled β times before an implementation decision. Qualitatively, it is a rate of information per sample. The allocation rules in §4 will make use of the KG factor when making a sampling decision at each stage of sampling. The KG factor uses the predictive distribution in (10) and the computational cost $c(\vec{x})$ of sampling at $\vec{x}$, measured by the computation time required.

Thus, the KG_β factor at time n for observing the value at a given singleton $x \in \{1, 2, \ldots, k\}$ is

$$\nu_n^{KG_\beta}(x) = V_n\left(x, A_n(x), \beta_n\right)/[\beta_n c(x)], \tag{15}$$

where β_n and $A_n(\cdot)$ may be chosen in an implementation-specific way (see §6.3). Similarly, the KG_β factor at time n for observing the difference in value between a pair of alternatives $\left(x^{(1)}, x^{(2)}\right)$ is

$$\nu_n^{KG_\beta}\left(x^{(1)}, x^{(2)}\right) = V_n\left(\left(x^{(1)}, x^{(2)}\right), A_n\left(x^{(1)}, x^{(2)}\right), \beta_n\right) / \left[\beta_n c\left(\left(x^{(1)}, x^{(2)}\right)\right)\right]. \tag{16}$$

If the computation time for a sample does not depend on $\vec{x}$, then $c(\vec{x}) = c|\vec{x}|$, where c is a positive constant cost per sample, and $|\vec{x}|$ is the length of $\vec{x}$. We adopt this model in numerical tests below.

4. Allocation Rules

This section discusses allocation rules, which use previous sampling information to decide how to take the next sample or samples, and which appear in Step 3 of the generic sampling algorithm in §2. The allocation rules discussed all search over a set of possible sampling decisions to find the one with the largest KG_β factor, but differ in the way in which this search is performed.

Let $\Xi = \{1, 2, \ldots, k\} \cup \left\{\left(x^{(1)}, x^{(2)}\right) \in \{1, 2, \ldots, k\}^2 : x^{(1)} \neq x^{(2)}\right\}$ denote the set of all singletons and pairs. For each allocation rule below, we let $\Xi_n \subseteq \Xi$ denote a possibly smaller set, and at each

iteration n, the allocation rule selects the sampling decision that maximizes the KG_β factor from §3.3 over this set,

$$\vec{x}_n = \arg\max_{\vec{x}\in\Xi_n} \nu_n^{\mathrm{KG}_\beta}(\vec{x}). \tag{17}$$

Certain ways of choosing the Ξ_n will be shown to improve the computation time of the algorithm while retaining theoretical convergence guarantees (in §5) and good empirical performance (in §7).

When calculating the KG_β factor $\nu_n^{\mathrm{KG}_\beta}(\vec{x})$, we replace strictly negative entries in the sampling covariance matrix Λ by 0, because Lemma 1 shows that this generates a larger VOI and corresponding KG_β factor. Then, if a pair of alternatives whose sampling covariance was replaced by 0 is selected for simulation by our allocation rule, we use independent sampling rather than CRN to simulate these alternatives. Otherwise, we use CRN when sampling pairs.

The expression (17) depends upon the choice for Ξ_n, and implicitly on the choice of β_n and $A_n(\vec{x})$ used to calculate $\nu_n^{\mathrm{KG}_\beta}(\vec{x})$. Thus, different allocation rules are specified by different methods for choosing Ξ_n, β_n, and $A_n(\vec{x})$. We define a class of allocation rules, called $\mathbf{KG}_\beta^2$ **allocation rules**, to be any that includes at least one singleton and one pair of alternatives in Ξ_n, and includes both $x^{(1)}$ and $x^{(2)}$ in $A_n(\vec{x})$, if $\vec{x} = \left(x^{(1)}, x^{(2)}\right)$, for each n. Within this larger class, we now define two more specific types of KG_β^2 allocation rules, which place additional conditions on Ξ_n.

An **idealized** KG_β^2 allocation rule (proposed in Frazier et al. 2011) is one in which $\Xi_n = \Xi$ for each n. Thus, an idealized KG_β^2 allocation rule looks over all the singleton and pairwise-difference KG_β factors and finds the largest one. A specific instance of an idealized KG_β^2 allocation rule would require specifying a choice for β_n and $A_n(\vec{x})$.

When k is large, the exhaustive maximization performed by an idealized KG_β^2 rule is too computationally intensive. Frazier et al. (2011) proposed an alternative to this exhaustive maximization, which checks only singletons and a subset of pairs of alternatives, but even that approach is too computationally intensive when $k \gg 10^3$ and is not easily amenable to theoretical analysis.

To allow for better performance in large problems in a way that also supports theoretical analysis, we propose here a new class of KG_β^2 allocation rules, called **accelerated** KG_β^2 allocation rules, which can be used when the alternatives are embedded in an integer lattice, or some other space that supports local search. An accelerated KG_β^2 allocation rule is one that chooses at least one singleton from $\{1, 2, \ldots, k\}$ and at least one pair from $\left\{\left(x^{(1)}, x^{(2)}\right) \in \{1, 2, \ldots, k\}^2 : x^{(1)} \neq x^{(2)}\right\}$, adding these to Ξ_n. Then, starting at each chosen singleton or pair $\vec{x}$, it applies a function $\tilde{f}$, which we call a "local search function", to produce a point $\tilde{f}(\vec{x})$, and adds this point to Ξ_n as well. The singleton or pair in Ξ_n with the best KG factor is then selected for evaluation, according to (17). The function $\tilde{f}$ can be defined in an implementation specific way, but would usually be designed to find a local optimum of the KG factor in the neighborhood of the passed input sampling decision.

Thus, an accelerated KG^2_β allocation rule is specified by a rule for choosing the starting singletons and pairs, and for β_n, $A_n(\vec{x})$, and $\tilde{f}$. One choice for $\tilde{f}$, implemented using a gradient-based local search appropriate for alternatives corresponding to an integer lattice, is provided in §6.3. Another choice, the identity map, $\tilde{f}(\vec{x}) = \vec{x}$, results in a form of random search.

The class of **KG_β allocation rules** is defined analogously to the class of KG^2_β allocation rules, except that only singletons (not pairs) may be sampled. That is, $\Xi_n \subseteq \{1, 2, \ldots, k\}$ in (17) for KG_β allocation rules. The notions of idealized and accelerated KG_β allocation rules are defined as for the KG^2_β allocation rules above, except that pairs are not included in the search. When $\Xi_n = \{1, \ldots, k\}$, $A_n(\vec{x}) = \{1, \ldots, k\}$, and $\beta_n = 1$, we recover the allocation rule proposed in Frazier et al. (2009).

5. Convergence Properties

This section shows that the generic sampling algorithm from §2, when used with known Λ, μ_0, and Σ_0, and with a KG^2_β allocation rule from §4 satisfying mild conditions, samples every alternative infinitely often, so that we learn the value of every alternative, and are able to find a global maximum $x^* \in \arg\max_x \theta(x)$ almost surely in the limit as the number of samples grows without bound. Frazier et al. (2009) proved these consistency results for the idealized KG_β algorithm with $A_n(\vec{x}) = \{1, \ldots, k\}$ and $\beta_n = 1$, and so the results here can be viewed as a generalization to KG^2_β and to algorithms that do not require exhaustive optimization over all alternatives. The presence of sampling correlations, however, require substantially different proof techniques from those used in Frazier et al. (2009).

These results depend up two assumptions and a condition, which are stated precisely below. The first assumption states that we require the parameters governing Λ, μ_0, and Σ_0 to be known and fixed. The second assumption states that there is genuine uncertainty about each alternative's performance. The condition restricts the choice of KG^2_β allocation rule, and is satisfied by the idealized KG^2_β and accelerated KG^2_β allocation rules from §4 as long as every $\vec{x} \in \Xi$ is chosen as a starting point for the local search infinitely often, with probability one.

ASSUMPTION 1. μ_0, Σ_0 and Λ are known.

ASSUMPTION 2. Σ_0 and Λ are positive definite.

CONDITION 1. Each $\vec{x} \in \Xi$ is included in Ξ_n infinitely often, with probability 1.

We now state our main result: that we become certain of the vector of true means θ eventually, as the conditional variance $\Sigma_n(x, x)$ of $\theta(x)$ converges to 0, and the conditional mean $\mu_n(x)$ converges to $\theta(x)$, for each x; and that the implementation decision that would be chosen if sampling stopped at time n, $\arg\max_x \mu_n(x)$, is eventually globally optimal. The proof may be found in Appendix A.

THEOREM 1. If Assumptions 1 and 2 hold, and if sampling occurs according to a KG^2_β allocation rule satisfying Condition 1, then: $\lim_{n\to\infty} \Sigma_n(x, x) = 0$ almost surely for each x; $\lim_{n\to\infty} \mu_n(x) = \theta(x)$ almost surely and in L^2 for each x; and $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely.

6. Implementation Features and Practicalities

This section discusses practical implementation choices arising in the generic algorithm in §2 and allocation rules in §4. This includes the specification of the functional form of the prior distribution and structure of the initial stage of sampling in Step 1, the empirical Bayes estimator used to assess μ_0, Σ_0 and Λ in Step 2, the choice of $A_n(\vec{x})$ and β_n used in KG^2_β allocation rules, and derivations of the gradients of the VOI and KG factors used by accelerated KG^2_β allocation rules. The convergence results in §5 do not depend on how these implementation issues are addressed, as long as Assumptions 1, 2 and Condition 1 are valid.

Several of the implementation choices discussed assume that the k alternatives may be represented as elements in a lattice in $\mathbb{Z}^d$. For example, in a manufacturing problem, there may be d decision variables, each of which represents the number of resources (machines, employees with given skill sets, etc.) that combine to define a specific alternative manufacturing system design. That is, for any alternative x, we can specify its grid coordinates $\{\zeta_i(x)\}_{i=1}^d$.

6.1. Functional Form of the Prior Distribution and Sampling Covariance (Step 1 of Generic Sampling Algorithm)

Step 1 of the generic sampling algorithm requires specification of the functional form of the sampling covariance and prior distribution for the unknown means, either fully, or more frequently in terms of parameters to be estimated later in Step 2. We discuss this choice here.

The functional form of the sampling covariance Λ is considered first. While several different forms are possible, we assume compound sphericity for simplicity. The compound sphericity assumption means that Λ can be specified with exactly two parameters: a common sampling variance σ^2_ϵ on the diagonals and a common sampling correlation across any pair of alternatives, ρ. All off-diagonal elements of Λ are the same. While the compound sphericity assumption is strong, it has been used by others to model the effect of CRN (Schruben and Margolin 1978, Tew and Wilson 1992), including in the context of CRN with kriging (Chen et al. 2012).

We now discuss the functional form of the prior distribution for the unknown means. When the alternatives may be embedded in a lattice, there may be a belief that the performance of two alternatives that are 'near' each other in this lattice are more likely to be similar than the performance of two alternatives that are 'distant' from each other. This motivates the notion that the prior distribution may be a multivariate normal distribution under which the covariance between the values of any two alternatives is a decreasing function of their distance from each other on the lattice. This is analogous to covariance functions used in GP priors over continuous functions. Inspired by this link to GP priors, we adopt the commonly used Gaussian kernel.

$$\Sigma_0(x,x') = \sigma_0^2 \exp\left\{-\sum_{i=1}^{d} \alpha_i \left[\zeta_i(x) - \zeta_i(x')\right]^2\right\}. \tag{18}$$

Here σ_0^2 is the homogeneous prior variance of the unknown means and $\vec{\alpha} = \{\alpha_i\}_1^d$ is a vector of scaling parameters. We also let η be a parameter for the mean in this model and let $\vec{1}$ be a vector of k ones, so that (18) and $\mu_0 = \eta\vec{1}$ define the prior distribution in (1).

Specification of the prior distribution parameters μ_0, Σ_0 can therefore be accomplished by specifying σ_0^2, $\vec{\alpha}$, and η. Kernels other than that in (18) would be handled similarly.

6.2. Initial Stage of Sampling (Step 1 of Generic Sampling Algorithm) and Empirical Bayes Parameter Update (Step 2 of Generic Sampling Algorithm)

Here we discuss the initial stage of sampling performed in Step 1, and the periodic empirical Bayes updates performed in Step 2 of the generic sampling algorithm. These steps are used when Λ, μ_0 or Σ_0 or some parameters of their functional forms are unknown, and require some estimation.

If an initial stage of sampling is required, we randomly select a set $\vec{x}_{01}$ of N_1 alternatives, sample once from each of them using CRN, sort them in descending order, and then take another sample from each of the first N_2 alternatives, denoted by the vector $\vec{x}_{02}$, using CRN ($N_2 < N_1$). We initialize the number of stages sampled so far to be $n = 2$ (one for each use of CRN), $\mathcal{X}_2$ to be the row concatenation of $\vec{x}_{01}$ and $\vec{x}_{02}$, and $\mathcal{Y}_2$ to be the outputs at those alternatives.

Once this initialization stage of samples is complete, and also periodically thereafter according to a fixed schedule, we estimate the parameters determining μ_0, Σ_0, and Λ in Step 2 of the generic sampling algorithm using a maximum likelihood estimator (MLE). Appendix B derives a MLE assuming that μ_0, Σ_0, and Λ take the functional form specified in §6.1, which has parameters σ_0^2, $\vec{\alpha}$, η, σ_ϵ^2 and ρ. This use of maximum likelihood estimation to estimate parameters within a Bayesian model is known as an empirical Bayes approach, and is common in GP regression. Relaxing the compound sphericity assumption or using a different GP prior in our proposed algorithm simply involves providing an alternative MLE for Λ, μ_0 and Σ_0.

We let N_3 denote the set of times at which the MLE will be performed, so N_3 contains $N_1 + N_2$. If computation time for the allocation rule is unimportant (e.g., because the simulations themselves are very time-consuming), one may perform the MLE before each new stage of sampling, in which case $N_3 = \{N_1 + N_2, N_1 + N_2 + 1, N_1 + N_2 + 2, \ldots\}$. In other situations, because computation of the MLE may be time-consuming, it may be beneficial to avoid recomputing the MLE at every stage. In our implementation, we update the MLE more frequently at first when additional samples tend to have more impact on parameter estimates, and then less frequently as more samples are acquired. If the parameters are known, we may skip these updates by setting $N_3 = \emptyset$.

6.3. Local Search Function and Other Implementation Choices in KG Allocation Rules (Step 3 of Generic Sampling Algorithm)

This section discusses implementation-specific choices for $A_n(\vec{x})$ and β_n in idealized and accelerated KG_β and KG_β^2 allocation rules. Additionally, for accelerated allocation rules, it discusses the choice of Ξ_n and the local search function $\tilde{f}$.

Except where otherwise noted in our numerical experiments, we set $\beta_n = 1$ and we chose $A_n(\vec{x})$ to be the alternatives in $\vec{x}$ and the best other sampled alternative given the observations available. So, for singletons $\vec{x} = (x)$, we set $A_n(x) = \{x, x_*\}$, where $x_* = \arg\max_{x' \in \mathcal{X}_n \setminus \{x\}} \mu_n(x')$. For pairs, $\vec{x} = (x^{(1)}, x^{(2)})$, we set $A_n(\vec{x}) = \left\{x^{(1)}, x^{(2)}, x_*\right\}$, where $x_* = \arg\max_{x' \in \mathcal{X}_n \setminus \{x^{(1)}, x^{(2)}\}} \mu_n(x')$.

We now describe the choice of Ξ_n used within accelerated KG^2_β and KG_β allocation rules in our numerical experiments. Denote the best and second best alternative (in terms of posterior mean) after n samples as

$$x_{n,b} = \arg\max_{x \in \mathcal{X}_n} \mu_n(x), \qquad x_{n,s} = \arg\max_{x \in \mathcal{X}_n \setminus \{x_{n,b}\}} \mu_n(x).$$

In accelerated KG_β allocation rules, eligible sampling decisions Ξ_n were $x_{n,b}$, $x_{n,s}$, a randomly chosen singleton, and the values of $\tilde{f}$ applied to those three sampling decisions. In accelerated KG^2_β allocation rules, eligible sampling decisions Ξ_n were $x_{n,b}$, a random singleton, $(x_{n,b}, x_{n,s})$, a random pair of alternatives, and the values of $\tilde{f}$ applied to those five sampling decisions.

In first stages of sampling, $\mathcal{X}_n$ may have too few elements for x_*, $x_{n,b}$ or $x_{n,s}$ to be defined. In such a case, a random sampling decision is used instead.

We now describe the local search function $\tilde{f}$ used within accelerated KG^2_β and KG_β allocation rules. This local search function assumes that the alternatives correspond to points on a grid embedded in a continuous space, as discussed in the beginning of §6, and also assumes that the prior is of the form specified in §6.1. This structure allows us to determine the gradient of the KG factors, and to use the gradient to locally optimize the KG factor in a neighborhood of $\vec{x}$, where $\vec{x}$ is interpreted as varying continuously. We round that local optimum to the nearest feasible grid point to obtain $\tilde{f}(\vec{x})$.

We first derive the gradient of the VOI, as it is required to determine the gradient of the KG factor. Specifically, we assess the gradient of $V_n(x, A_n(x), \beta)$ in $\mathbb{R}^d$ and of $V_n\left((x^{(1)}, x^{(2)}), A_n\left(x^{(1)}, x^{(2)}\right), \beta\right)$ in $\mathbb{R}^{2d}$, where $A_n(\vec{x})$ is as described above.

We abuse notation slightly by writing the gradient of $V_n(x, A_n(x), \beta)$ in $\mathbb{R}^d$ in terms of derivatives with respect to the d coordinates of x rather than with respect to the $\zeta_i(x)$, in order to simplify notation. Similarly, for pairs $\vec{x}$, we write the gradient of $V_n(\vec{x}, A_n(\vec{x}), \beta)$ in $\mathbb{R}^{2d}$ by referring directly to the alternatives $\vec{x}$ rather than indirectly through the function ζ_i that embeds them in the grid.

First consider the case of the singleton $\vec{x} = x$. Recall that $V_n(x, A_n(x), \beta) = sf(-\Delta/s)$, where $\Delta = |\mu_n(x) - \mu_n(x_*)|, s = |\widetilde{\sigma}_n(x, x, \beta) - \widetilde{\sigma}_n(x, x_*, \beta)|$, and $f(z) = \varphi(z) + z\Phi(z)$. Direct calculation then reveals that

$$\begin{aligned}\nabla_x [V_n(x, A_n(x), \beta)] = \varphi(\Delta/s) \cdot \operatorname{sign}[\widetilde{\sigma}_n(x, x, \beta) - \widetilde{\sigma}_n(x, x_*, \beta)] \cdot \nabla_x [\widetilde{\sigma}_n(x, x, \beta) - \widetilde{\sigma}_n(x, x_*, \beta)] \\ - \Phi(-\Delta/s) \cdot \operatorname{sign}[\mu_n(x) - \mu_n(x_*)] \cdot \nabla_x [\mu_n(x) - \mu_n(x_*)].\end{aligned} \tag{19}$$

Detailed derivations of $\nabla_x [\mu_n(x')]$ and $\nabla_x [\widetilde{\sigma}_n(x, x', \beta)]$ for arbitrary x' are given in Appendix C.

Second, consider the case of the pair $\vec{x} = \left(x^{(1)}, x^{(2)}\right)$. Letting $a = \mu_n(A_n(\vec{x}))$ and $b = \widetilde{\sigma}(\vec{x}, A_n(\vec{x}), \beta)$ we have from §3.2 that

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = h(a,b).$$

To support taking the derivative of this quantity, we now recall Algorithms 1 and 2 from Frazier et al. (2009) for computing $h(a,b) = E[\max_i a(i) + b(i)Z] - \max_i a(i)$. We first reorder the components of a and b so that the $b(i)$ are in non-decreasing order and ties in b are broken so that $a(i) \leq a(i+1)$ if $b(i) = b(i+1)$. Then, we remove all those entries i for which $a(i) + b(i)z < \max_{j \neq i} a(j) + b(j)z$ for all values of z (this is accomplished by Algorithm 1 in Frazier et al. (2009)). This gives new vectors a' and b' with $|a'| = |b'| \leq |a| = |b|$. Set $\gamma(i) = \frac{a'(i+1) - a'(i)}{b'(i+1) - b'(i)}$ for $i = 1, 2, \ldots, |a'| - 1$. Then

$$V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) = h(a,b) = \sum_{i=1}^{|a'|-1} \left[b'(i+1) - b'(i)\right] f\left(-\left|\gamma(i)\right|\right)$$

if $|a'| > 1$ and the sum is taken to be 0 if $|a'| = 1$. Computation then reveals that

$$\begin{aligned} \nabla_{\vec{x}}\left[V_n\left(\vec{x}, A_n(\vec{x}), \beta\right)\right] = \sum_{i=1}^{|a'|-1} & \varphi\left(\gamma(i)\right) \nabla_{\vec{x}}[b'(i+1) - b'(i)] \\ & -\Phi\left(-|\gamma(i)|\right) \operatorname{sign}[a'(i+1) - a'(i)] \nabla_{\vec{x}}[a'(i+1) - a'(i)]. \end{aligned} \tag{20}$$

For each i, $a'(i)$ and $b'(i)$ are equal to $a(j)$ and $b(j)$ for j given by the reordering procedure above, and $a(j)$ and $b(j)$ are the jth components of $a = \mu_n(A_n(\vec{x}))$ and $b = \widetilde{\sigma}(\vec{x}, A_n(\vec{x}), \beta)$ respectively. Thus, $\nabla_{\vec{x}}[a'(i)]$ and $\nabla_{\vec{x}}[b'(i)]$ are equal to $\nabla_{\vec{x}}\left[\mu_n\left(x'\right)\right]$ and $\nabla_{\vec{x}}\left[\widetilde{\sigma}_n\left(\vec{x}, x', \beta\right)\right]$, where x' is the jth element in $A_n(\vec{x})$. Derivations of these quantities are given in Appendix C.

We now consider the gradient of the KG_β factors in $\mathbb{R}^d$. Recalling (15) and (16), we have

$$\nabla_{\vec{x}}\left[\nu_n^{KG_\beta}(\vec{x})\right] = \left(\nabla_{\vec{x}}\left[V_n\left(\vec{x}, A_n(\vec{x}), \beta\right)\right] \cdot c(\vec{x}) - V_n\left(\vec{x}, A_n(\vec{x}), \beta\right) \cdot \nabla_{\vec{x}}\left[c(\vec{x})\right]\right) / \left(\beta[c(\vec{x})]^2\right) \tag{21}$$

for $\vec{x} = x$ or $\left(x^{(1)}, x^{(2)}\right)$. In the case of homogeneous sampling costs for each alternative ($c(\vec{x}) = c|\vec{x}|$), we have $\nabla_{\vec{x}}\left[c(\vec{x})\right] = 0$. Hence (21) is determined by preceding results as

$$\nabla_{\vec{x}}\left[\nu_n^{KG_\beta}(\vec{x})\right] = \nabla_{\vec{x}}\left[V_n\left(\vec{x}, A_n(\vec{x}), \beta\right)\right] / \left(\beta c(\vec{x})\right).$$

7. Numerical Results

Beyond asymptotic convergence to the optimal solution, we are interested in the rate in which solutions improve for even small numbers of samples. We measure this performance by the expected opportunity cost (the difference between the true best and the estimated best $x_{n,b}$, as defined in §6.3, at each time n), $\mathbb{E}\left[\max_x \theta_x - \theta_{x_{n,b}}\right]$.

In this section we present numerical results to explore the behavior of the proposed algorithm, allocations rules, and implementation choices from §6 in order to answer the following questions.

Does pairwise sampling with CRN provide an efficiency benefit, even if approximations are made to simplify computations? How much benefit can the KG and KG^2 allocation rules give, on problems with combinatorially large numbers of solutions, as compared to other benchmark algorithms such as a random search which is enhanced with a Gaussian process metamodel (which we call RSGP and describe below) and Industrial Strength COMPASS (Xu et al. 2010).

Except as noted below, the KG and KG^2 allocation rules used the Gaussian process prior for unknown means, compound sphericity assumption for samples, MLE and empirical Bayes estimation, and other parameters as described in §6. When μ_0, Σ_0 and Λ were not known, the parameters for the initial stage of sampling were $N_1 = 10d, d \leq N_2 \leq 2d$, where d is the dimension of the problem, and we let N_3 contain $N_1 + N_2$ and stage numbers that allowed the period between updates to increase from 30 to 60 as sampling continued. In cases where sampling was done without CRN, we performed maximum likelihood estimation with ρ fixed to 0.

7.1. How do the approximations interact?

This section assesses the relative importance of several features and approximations described above: the allocation rule, approximations due to accelerated allocations and parameter estimation, and deviations from the assumed sampling correlation structure under CRN. Specifically, we assess the $12 = 2 \times 3 \times 2$ combinations that result from combining each level of the following three factors:
Allocation: KG_β allocation rule (no CRN); or KG^2_β allocation rule (CRN allowed).
Approximation: Idealized allocation rule with known parameters; accelerated allocation rule with known parameters; or accelerated allocation with unknown parameters $(\sigma_0^2, \vec{\alpha}, \eta, \sigma_\epsilon^2, \rho)$ fit as in §6.2.
Sampling with CRN: Samples satisfy compound sphericity (with $\rho(i,j) = 0.25$ for $i \neq j$); or decreasing correlations (with $\rho(i,j) = \exp[-(i-j)^2/50]$ for $i \neq j$) even though compound sphericity may be (incorrectly) assumed by the parameter fitting.

We do so for randomly generated problem instances with a small (100) number of alternatives. We generate 500 problem instances. In each problem, the 100 alternatives had means distributed as a $\mathcal{N}(\mu_0, \Sigma_0)$ with $\mu_0 = \vec{0}$ and $\Sigma_0(i,j) = 100\exp[-(i-j)^2/50]$ for $i, j = 1, 2, \ldots, 100$. We assumed a homogeneous sampling variance $\sigma_\epsilon^2 = 50$. We set $\beta = 1$ and so refer to KG_1 and KG_1^2.

Figure 1 shows the expected opportunity cost of a potentially incorrect selection, on a logarithmic scale, as a function of the total number of samples. The maximum size of the 95% confidence intervals is 0.28 at sample size 100, 0.09 at sample size 300, and 0.05 at sample size 500.

Not surprisingly, idealized KG allocation rules performed better than their accelerated counterparts (the idealized does an exhaustive, not local, search to maximize the KG factor). The degree of sub-optimality was not particularly great in any setting where parameters were known. A greater degree of sub-optimality was seen when parameter estimation was used. The deterioration due

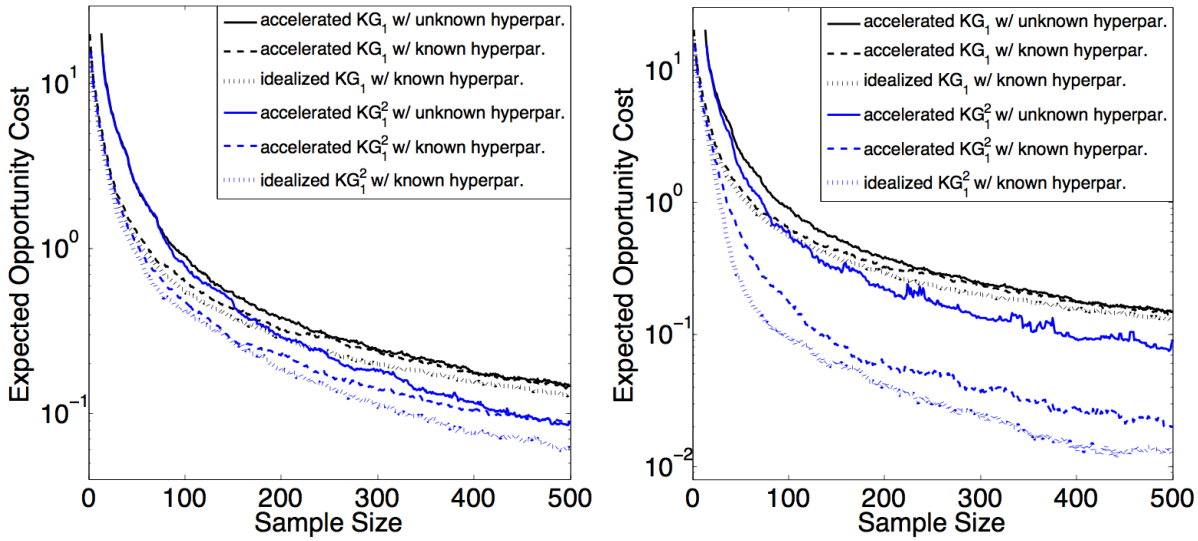

Figure 1 **Performance of selected algorithms in the grid test problem with compound sphericity (left plot) and decreasing correlations (right plot).**

to parameter estimation was not significant for the KG_1 allocation, even when sphericity did not apply and parameters were (incorrectly) estimated with the sphericity assumption (right panel, top three lines). The degradation in performance due to parameter estimation with the KG_1^2 allocation was not too significant when sphericity was correctly assumed (left panel, bottom three curves).

All else fixed, a KG^2 allocation with CRN improved upon the performance of its corresponding KG allocation with independent sampling. Thus, the ability of sampling pairs with CRN offered an important benefit beyond sampling only one alternative independently at a time (both panels).

Moreover, we observed that the accelerated KG_1^2 allocation rule, even when parameter estimation was used, performed better than the idealized KG_1 allocation, which had the advantage of 'knowing' the true sampling correlation and of doing an exhaustive search over KG factors. Thus, the benefit of CRN outweighed the penalties associated with sub-optimality in the accelerated KG_1^2 allocation rule with unknown parameters, once 200 samples were observed to get stable parameter estimates (even when sphericity was incorrectly assumed by the MLE, right panel).

In experiments not shown here for reasons of space, we found other interesting observations. One, when we set $\rho(i,j) = 0.5$ rather than $\rho(i,j) = 0.25$ for all $i \neq j$, the expected opportunity costs decreased. This is consistent with the benefit offered by sampling pairs being increasing in a (common) sampling correlation ρ. Two, we experimented with the number of randomly selected singletons and pairs that were included in Ξ_n for the accelerated allocations. Increasing that number to 2 or 3 provided a practical improvement in performance in the approximate KG and KG^2 allocations, but the benefit of adding random points beyond 4 or 5 had little marginal increase. Three,

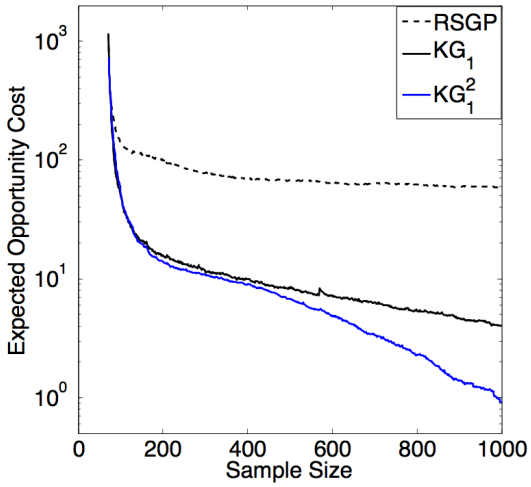

Figure 2 **Expected opportunity cost for a benchmark algorithm, (Random Search with Gaussian Processes, RSGP), accelerated KG_1 (KG_1) and accelerated KG_1^2 (KG_1^2) allocation rules as a function of the total number of samples (on discrete Rosenbrock function).**

experiments with several values of β_n for KG_β^2 allocation rules did not reveal a large difference in performance due to the choice of β_n.

7.2. Comparison with RSGP on a Rosenbrock Problem with 10^6 Alternatives

This section explores the performance of the procedures when there are a very large number of alternatives. The problem considered is a discretized version of a 6-dimensional Rosenbrock function with 10^6 alternatives. Each alternative x corresponds to a point in the grid with coordinates $\zeta(x) = \{\zeta_i(x)\}_{i=1}^6 \in [-0.8, -0.5, \ldots, 1.9]^6$ and has value

$$\theta(x) = -\sum_{i=1}^{5} \left[100\left[\zeta_i(x)^2 - \zeta_{i+1}(x)\right]^2 + \left[\zeta_i(x) - 1\right]^2\right].$$

The computation required for idealized KG allocation rules is not practical when there are such a large number of alternatives. This section assesses differences in performance between the accelerated KG_1 and accelerated KG_1^2 allocation rules, as well as a benchmark algorithm that we introduce, called RSGP. The RSGP samples uniformly at random and uses the Gaussian Process model and parameter estimation tools in §6 to estimate the performance for each alternative by its posterior mean when selecting the best alternative.

The sampling noise satisfies the compound sphericity assumption, with $\sigma_\epsilon^2 = 125$ and $\rho(i,j) = 0.4$ for all $i \neq j$. These values were assumed unknown in this test, and the empirical Bayes approach described in §6.2 was used to estimate the GP prior and sampling covariance.

Figure 2 shows the opportunity cost, averaged over 200 sample paths, of the accelerated KG_1 and accelerated KG_1^2 allocation rules, and the benchmark RSGP. Both KG allocation rules dramatically

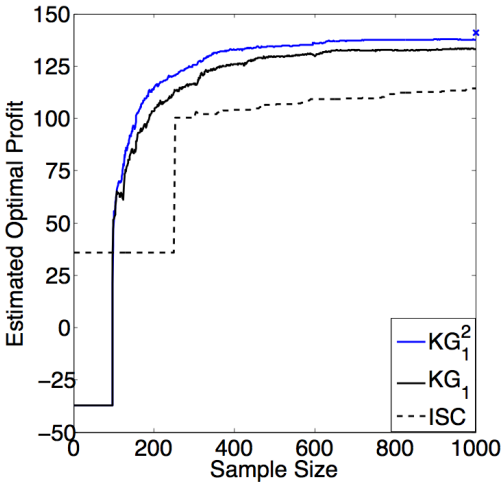

Figure 3 **Performance of the accelerated KG_1^2 and accelerated KG_1 allocation rules, and Industrial Strength COMPASS (ISC) for the assemble-to-order (ATO) problem.**

outperformed RSGP. This is because the KG factors steered sampling to areas that more efficiently identified local extrema. The KG_1 and KG_1^2 performed similarly through about 500 samples, but KG_1^2 provided better solutions thereafter. Exploring sample paths indicates that this was because both KG_1 and KG_1^2 initially identified regions of good local extrema, which occurred at about the same rate. Then, when good local extrema were found, the use of CRN helped KG_1^2 find better solutions more quickly, as compared to KG_1, near such local extrema.

7.3. Comparison with ISC on the Assemble to Order Problem

We now compare the accelerated KG_1 and KG_1^2 allocation rules with a well-known algorithm, Industrial Strength COMPASS (ISC, developed by Xu et al. 2010). We do so for the Assemble to Order (ATO) problem described in Hong et al. (2012), which is a variation on the problem studied by Hong and Nelson (2006), and has a combinatorially large number (21^8) of alternatives.

In the ATO problem, orders for 5 different products arrive according to independent Poisson processes with constant arrival rates. Products are made up of a collection of items of 8 different types. Items are either key items or non-key items. If any of the key items are out of stock then the product order is lost. If all key items are in stock, then the order is assembled from all key items and the available non-key items. Each item sold brings a profit, and each item in inventory incurs a holding cost per unit time. There is an inventory capacity 20 for each item. Items are produced one at a time on dedicated machines. The production time for each item is normally distributed, truncated at 0. The system operates under a continuous-review base stock policy under which each item k has a target base stock b_k, and each demand for an item triggers a replenishment order for that item. Each simulation replication starts from a fully stocked system with no orders in production, has a warm-up period of 20 time units, then captures statistics for the next 50 time

units of operation. The goal is to maximize the expected total profit per unit time by selecting the target inventory level vector $b = (b_1, b_2, \ldots, b_8)$. See Hong et al. (2012) for more details and code.

We calculate each algorithm's performance by collecting the true expected total profit (estimated in a post-processing step through exhaustive simulation) of the algorithm's current solution, as a function of the sample size. We then average this value over 100 independent sample paths for each algorithm. We fix the starting solution of KG_1 and KG_1^2 to the inventory capacity, and randomize the initial solution of ISC over the feasible set $\{b : 0 \leq b_k \leq 20, b_k \in \mathbb{Z}\}$. Thus, KG_1 and KG_1^2 were forced to start searching with a worse initial alternative to sample than did ISC, on average.

Figure 3 shows the average performance of the three algorithms. This average performance jumped when algorithms finished their initialization phases (Step 1 of the KG algorithms), which occurred at 250 samples for ISC and 95 samples for the two KG algorithms. The height of the x at the right edge of the plot (at $x = 1000$) gives the value (141, estimated through exhaustive simulation) of the best solution found by all sample paths across all three algorithms. This best solution was discovered by a KG algorithm. The true optimal solution is unknown. The accelerated KG_1^2 allocation rule outperformed the accelerated KG_1 allocation rule, which in turn outperformed ISC for this problem, in terms of achieving a higher quality solution with fewer samples. It is also important to consider the total amount of computation time required to reach a given solution quality. ISC required an average of 27 minutes of computation time to complete, taking 1084 samples on average. Its average profit upon completion was 115.53. To reach this same level of solution quality achieved by ISC, KG_1 took 279 samples on average and required 9 minutes of computation time, while KG_1^2 took 203 samples on average and required 5.5 minutes of computation time. The two KG algorithms required fewer samples and less computation time than did ISC, with KG_1^2 delivering additional efficiency above and beyond that delivered by KG_1.

While the KG algorithms outperformed ISC in terms of total computation time to reach a given level of solution quality on the ATO problems, algorithms like KG_β and KG_β^2 that rely on kriging or Gaussian-process regression may consume substantial computational resources in deciding where to sample, which may make them less suitable for problems in which simulation can be performed very quickly. When simulation samples come from a complex, long-running simulator, this is relatively unimportant, and algorithms like KG_β^2 that find good solutions in few samples also work well in terms of overall computation time.

Figure 4 shows the time taken in a single sample path of the KG_β^2 algorithm. It shows that the CPU time per sampling decision increased over time, with a baseline level of computation due to gradient-based optimization of the KG factor, and spikes at regular intervals due to the empirical Bayes update of parameters. These spikes, which are so prominent in the right-hand panel of Figure 4, would also be present in any algorithm using kriging with adaptively updated parameter

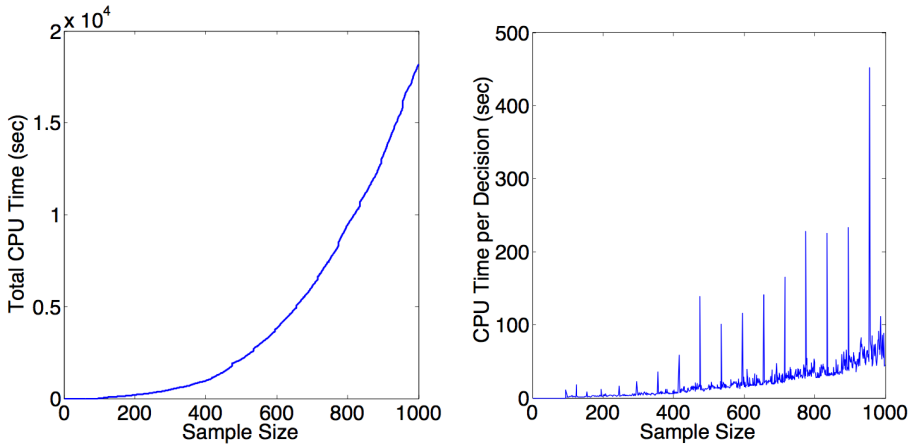

Figure 4 **CPU time spent in a sample path of KG_1^2, as a function of the sample size, on the ATO problem.**

estimates. The increase with sample size in both the time to perform gradient-based optimization of the KG factor, and to perform empirical Bayes updates, was due to the increasing size of the matrices being manipulated for maximum likelihood estimation and for kriging-based prediction.

These points suggest potential future research directions: how to balance frequency of parameter updates to improve performance with the cost of computing them; how to speed up and improve parameter estimation; adaptation or development of localized submodels for kriging approximation to reduce the number of samples included in local gradient search to optimize KG factors; how much time to spend on the local search (balancing some improvement versus perfect improvement in these steps). Related to this last point, we did derive and test second-order methods (not shown) to find local optimizers of the KG factors but found they did not give CPU cost per iteration benefits relative to Matlab's fminsearch and simple gradient search on some test problem.

In summary, our algorithms demonstrate superior efficiency compared to others in problems with large solution spaces and when samples are moderately to very computationally expensive.

8. Conclusions

We contributed to the area of discrete optimization via simulation, where the value of the best alternative is to be estimated by simulation, by developing a fully sequential algorithm based on new value of information tools. Those tools are able to take advantage of both correlated prior beliefs and correlated sampling distributions. We gave easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed. The implementation presented here takes advantage of machine learning tools that enable exploring combinatorially large solution spaces, with run times that are a low order polynomial in the number of samples observed (which is much better than a low order polynomial in the size of the solution space). We also derived 'accelerated' versions of the algorithms that use local search when alternatives can be embedded in

a continuous space. That acceleration takes advantage of gradient information about the Bayesian value of information, rather than the more common technique of using gradient information about the response surface, to improve practical performance. Numerical results show that there is a distinct benefit for being able to use both correlated prior beliefs and correlated sampling in simulation optimization using the Bayesian value of information framework.

Appendix

A. Mathematical Proofs

Proof of Lemma 1. From the definition of the function $h(\cdot,\cdot)$ (just after (14)), for any vectors a, b and b' with $b(i) \leq b'(i)$ for all i, we have $h(a,b) \leq h(a,b')$. From (14), we have $V_n(\vec{x},A,\beta) = h\left(\mu_n(A), \widetilde{\sigma}_n\left(\vec{x},A,\beta\right)\right)$, where for all $x' \in A$, the element of $\widetilde{\sigma}_n(\vec{x},A,\beta)$ corresponding to x' is $\left[\Sigma_n\left(x',x^{(1)}\right) - \Sigma_n\left(x',x^{(2)}\right)\right]/B$, where B is the denominator (in the lower equation for pairs) in (11). Hence we only need to show that B is a decreasing function of $\rho\left(x^{(1)},x^{(2)}\right)$. The result follows immediately by observing $\Lambda\left(x^{(1)},x^{(2)}\right) = \rho\left(x^{(1)},x^{(2)}\right)\left[\Lambda(x^{(1)},x^{(1)})\Lambda(x^{(2)},x^{(2)})\right]^{1/2}$. □

Preliminary results for the convergence proofs. We first state and prove several lemmas needed to prove the convergence results stated in §5. These lemmas all assume Assumptions 1 and 2. Condition 1 is assumed only in the proof of Theorem 1.

LEMMA 2. *There exist random variables $\mu_\infty \in \mathbb{R}^k$ and $\Sigma_\infty \in \Sigma_+^k$ (the space of $k \times k$ positive semi-definite matrices), such that μ_n converges to μ_∞, and Σ_n converges to Σ_∞ almost surely.*

Proof of Lemma 2. Let (μ_n, Σ_n) and $M_n = (\mu_n, \Sigma_n + \mu_n\mu_n^T)$. We can write the components of M_n as the conditional expectation of an integrable random variable with respect to $\mathcal{X}_n$, $\mathcal{Y}_n$ by $\mu_n = \mathbb{E}_n\theta, \Sigma_n + \mu_n\mu_n^T = \mathbb{E}_n\theta\theta^T$. This implies that M_n is a uniformly integrable martingale and hence converges almost surely (Doob's second martingale convergence theorem, e.g. see Oksendal 2003, App. C). Because (μ_n, Σ_n) is a continuous transformation of M_n, it also converges almost surely to some random variable $(\mu_\infty, \Sigma_\infty)$. □

LEMMA 3. $\Sigma_n(x',x) = \Sigma_0(x',x) - \Sigma_0\left(x,\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right)$.

Proof of Lemma 3. Let $i_{x'}$ be the index of x' in $\mathcal{X}_{n,x}$. (If x' appears more than once, let it be the index of one occurance.) Let $e_{x'}$ be a column vector with length $|\mathcal{X}_n|+1$ that has value 1 at entry $i_{x'}$ and 0 elsewhere. Let e_x be defined similarly. Using (5), (7) and the symmetry of $[S_n]^{-1}$, we then have

$$\begin{aligned}
\Sigma_n(x',x) &= e_{x'}^T\Sigma_n\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)e_x = e_{x'}^T\left(I_{|\mathcal{X}_n|+1} - K_n(x)\left[I_{|\mathcal{X}_n|},\vec{0}\right]\right)\Sigma_0\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)e_x \\
&= e_{x'}^T\Sigma_0\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)e_x - e_{x'}^TK_n(x)\left[I_{|\mathcal{X}_n|},\vec{0}\right]\Sigma_0\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)e_x \\
&= \Sigma_0(x',x) - e_{x'}^T\Sigma_0\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)\left[I_{|\mathcal{X}_n|},\vec{0}\right]^T[S_n]^{-1}\left[I_{|\mathcal{X}_n|},\vec{0}\right]\Sigma_0\left(\mathcal{X}_{n,x},\mathcal{X}_{n,x}\right)e_x \\
&= \Sigma_0(x',x) - \Sigma_0\left(x',\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x\right) = \Sigma_0(x',x) - \Sigma_0\left(x,\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right). \quad \square
\end{aligned}$$

LEMMA 4. $\Sigma_{n+1}(x,x) \leq \Sigma_n(x,x)$ *for all* x.

Proof of Lemma 4. Using standard results from Bayesian linear regression (e.g., Gelman et al. 2004, Sec. 14.6) and the Sherman-Morrison-Woodbury formula (e.g., Rasmussen and Williams 2006, App. A.3), the posterior variance Σ_{n+1} of θ can be computed recursively by

$$\Sigma_{n+1} = \Sigma_n - \Sigma_n X_{n+1} \left[X_{n+1}^T (\Lambda + \Sigma_n) X_{n+1} \right]^{-1} X_{n+1}^T \Sigma_n,$$

where

$$X_{n+1} = \begin{cases} e_x, & \text{if } \vec{x}_{n+1} = x, \\ [e_{x^{(1)}}, e_{x^{(2)}}], & \text{if } \vec{x}_{n+1} = \left(x^{(1)}, x^{(2)}\right), \end{cases}$$

and e_x is a $k \times 1$ vector with a value of 1 at the entry for x and 0 elsewhere.

It is clear that $\Lambda + \Sigma_n$ and $X_{n+1}^T (\Lambda + \Sigma_n) X_{n+1}$ are positive definite. Hence for any x,

$$\Sigma_{n+1}(x,x) = \Sigma_n(x,x) - e_x^T \Sigma_n X_{n+1} \left[X_{n+1}^T (\Lambda + \Sigma_n) X_{n+1} \right]^{-1} X_{n+1}^T \Sigma_n e_x \le \Sigma_n(x,x). \quad \square$$

LEMMA 5. *For all x and $x^{(1)} \neq x^{(2)}$, $P\left(x^{(1)}, x^{(2)}\right) = \Lambda\left(x^{(1)}, x^{(1)}\right) + \Lambda\left(x^{(2)}, x^{(2)}\right) - 2\Lambda\left(x^{(1)}, x^{(2)}\right) > 0$ and*

$$\nu_n^{\mathrm{KG}_\beta}(x) \le \frac{1}{c(x)} \sqrt{\frac{2 \max_{x'} \Sigma_0(x',x') \Sigma_n(x,x)}{\beta \pi \Lambda(x,x)}},$$

$$\nu_n^{\mathrm{KG}_\beta}\left(x^{(1)}, x^{(2)}\right) \le \frac{1}{c\left(x^{(1)}, x^{(2)}\right)} \sqrt{\frac{2 \max_{x'} \Sigma_0(x',x')}{\beta \pi P\left(x^{(1)}, x^{(2)}\right)}} \left[\sqrt{\Sigma_n\left(x^{(1)}, x^{(1)}\right)} + \sqrt{\Sigma_n\left(x^{(2)}, x^{(2)}\right)} \right].$$

Proof of Lemma 5. First, we have

$$\begin{aligned} V_n(\vec{x}, A_n(\vec{x}), \beta) &= \mathbb{E}_n \left[\max \mu_{n+1}(A_n(\vec{x})) \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta \right] - \max \mu_n(A_n(\vec{x})) \\ &= \mathbb{E}\left[\max \left\{ \mu_n(A_n(\vec{x})) + \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta) Z \right\} \right] - \max \mu_n(A_n(\vec{x})) \\ &\le \max \mu_n(A_n(\vec{x})) + \mathbb{E}\left[\max \left\{ \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta) Z \right\} \right] - \max \mu_n(A_n(\vec{x})) \\ &= \mathbb{E}\left[\max \left\{ \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta) Z \right\} \right] \le \mathbb{E}\left[\max \left\{ |\widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| \cdot |Z| \right\} \right] \\ &= \mathbb{E}|Z| \cdot \max \left\{ |\widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| \right\} = \sqrt{2/\pi} \cdot \max \left\{ |\widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| \right\} \\ &= \sqrt{2/\pi} \cdot \max_{j=1,2,\ldots,|A_n(\vec{x})|} |e_j^T \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)|, \end{aligned}$$

where e_j is a $|A_n(\vec{x})| \times 1$ vector with 1 at entry j and 0 elsewhere.

We now derive an upper bound on $e_j^T \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)$. First, $P\left(x^{(1)}, x^{(2)}\right) = [e_{x^{(1)}} - e_{x^{(2)}}]^T \Lambda [e_{x^{(1)}} - e_{x^{(2)}}] > 0$ because Λ is positive definite by Assumption 2, where $e_{x^{(j)}}$ is a vector with 1 at entry $x^{(j)}$ and 0 elsewhere. Similarly, we have $\Sigma_n\left(x^{(1)}, x^{(1)}\right) + \Sigma_n\left(x^{(2)}, x^{(2)}\right) - 2\Sigma_n\left(x^{(1)}, x^{(2)}\right) \ge 0$ because Σ_n is positive semi-definite. Now applying (11) and Lemma 4, for $\vec{x} = x$ we have

$$\begin{aligned} |e_j^T \widetilde{\sigma}_n(x, A_n(x), \beta)| &= \frac{|e_j^T \Sigma_n(A_n(x), x)|}{\sqrt{\beta^{-1}\Lambda(x,x) + \Sigma_n(x,x)}} = \frac{|\Sigma_n\left(A_n^{(j)}(x), x\right)|}{\sqrt{\beta^{-1}\Lambda(x,x) + \Sigma_n(x,x)}} \\ &\le \sqrt{\frac{\Sigma_n\left(A_n^{(j)}(x), A_n^{(j)}(x)\right) \Sigma_n(x,x)}{\beta^{-1}\Lambda(x,x)}} \le \sqrt{\frac{\Sigma_0\left(A_n^{(j)}(x), A_n^{(j)}(x)\right) \Sigma_n(x,x)}{\beta^{-1}\Lambda(x,x)}}, \end{aligned}$$

where $A_n^{(j)}(\vec{x})$ is the jth component of $A_n(\vec{x})$. Similarly for $\vec{x} = \left(x^{(1)}, x^{(2)}\right)$ we have

$$|e_j^T \widetilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| = \frac{|e_j^T \Sigma_n\left(A_n(\vec{x}), x^{(1)}\right) - e_j^T \Sigma_n\left(A_n(\vec{x}), x^{(2)}\right)|}{\sqrt{\beta^{-1} P\left(x^{(1)}, x^{(2)}\right) + \Sigma_n\left(x^{(1)}, x^{(1)}\right) + \Sigma_n\left(x^{(2)}, x^{(2)}\right) - 2\Sigma_n\left(x^{(1)}, x^{(2)}\right)}}$$

$$\leq \frac{|\Sigma_n\left(A_n^{(j)}(\vec{x}),x^{(1)}\right)|+|\Sigma_n\left(A_n^{(j)}(\vec{x}),x^{(2)}\right)|}{\sqrt{\beta^{-1}P\left(x^{(1)},x^{(2)}\right)}}$$
$$\leq \sqrt{\frac{\Sigma_n\left(A_n^{(j)}(\vec{x}),A_n^{(j)}(\vec{x})\right)}{\beta^{-1}P\left(x^{(1)},x^{(2)}\right)}}\left[\sqrt{\Sigma_n\left(x^{(1)},x^{(1)}\right)}+\sqrt{\Sigma_n\left(x^{(2)},x^{(2)}\right)}\right]$$
$$\leq \sqrt{\frac{\Sigma_0\left(A_n^{(j)}(\vec{x}),A_n^{(j)}(\vec{x})\right)}{\beta^{-1}P\left(x^{(1)},x^{(2)}\right)}}\left[\sqrt{\Sigma_n\left(x^{(1)},x^{(1)}\right)}+\sqrt{\Sigma_n\left(x^{(2)},x^{(2)}\right)}\right].$$

The claimed bounds in the lemma for $\nu_n^{\mathrm{KG}_\beta}(x)$ and for $\nu_n^{\mathrm{KG}_\beta}\left(x^{(1)},x^{(2)}\right)$ follow directly. □

LEMMA 6. *Under the allocation rule $x_1=x_2=\cdots=x_n=x$, $\Sigma_n(x,x)$ decreases to 0 as $n\to+\infty$. Under the allocation rule $\vec{x}_1=\vec{x}_2=\cdots=\vec{x}_n=\left(x^{(1)},x^{(2)}\right)$, $\Sigma_n\left(x^{(1)},x^{(1)}\right)$ and $\Sigma_n\left(x^{(2)},x^{(2)}\right)$ decrease to 0 as $n\to+\infty$.*

Proof of Lemma 6. Lemma 4 shows that $\Sigma_n(x,x)$ is a decreasing sequence bounded below by zero, for all x. It suffices to show that the limit is 0 under these two cases.

First consider the case when $x_1=x_2=\cdots=x_n=x_0$. Note $\Sigma_0\left(\mathcal{X}_n,\mathcal{X}_n\right)=\Sigma_0(x_0,x_0)ee^T$, where e is an $n\times 1$ vector with n entries of 1. By Lemma 3 and the Sherman-Morrison-Woodbury formula, for any x and x',

$$\begin{aligned}\Sigma_n(x,x') &= \Sigma_0(x,x')-\Sigma_0(x,\mathcal{X}_n)\left[S_n\right]^{-1}\Sigma_0(\mathcal{X}_n,x')\\ &= \Sigma_0(x,x')-\Sigma_0(x,x_0)\Sigma_0(x',x_0)e^T\left[\Sigma_0(x_0,x_0)ee^T+\Lambda(x_0,x_0)I_n\right]^{-1}e\\ &= \Sigma_0(x,x')-\frac{\Sigma_0(x,x_0)\Sigma_0(x',x_0)}{\Lambda(x_0,x_0)}e^T\left[I_n-\frac{\Sigma_0(x_0,x_0)}{n\Sigma_0(x_0,x_0)+\Lambda(x_0,x_0)}ee^T\right]e\\ &= \Sigma_0(x,x')-\frac{n\Sigma_0(x,x_0)\Sigma_0(x',x_0)}{n\Sigma_0(x_0,x_0)+\Lambda(x_0,x_0)}.\end{aligned}$$

Specifically,

$$\Sigma_n(x_0,x_0)=\Sigma_0(x_0,x_0)\left[1-\frac{n\Sigma_0(x_0,x_0)}{n\Sigma_0(x_0,x_0)+\Lambda(x_0,x_0)}\right]\to 0 \text{ as } n\to+\infty.$$

Next consider the case when $\vec{x}_1=\vec{x}_2=\cdots=\vec{x}_n=\left(x_0^{(1)},x_0^{(2)}\right)$. Let

$$D_1=\begin{bmatrix}\Sigma_0\left(x_0^{(1)},x_0^{(1)}\right) & \Sigma_0\left(x_0^{(1)},x_0^{(2)}\right)\\ \Sigma_0\left(x_0^{(1)},x_0^{(2)}\right) & \Sigma_0\left(x_0^{(2)},x_0^{(2)}\right)\end{bmatrix},\qquad D_2=\begin{bmatrix}\Lambda\left(x_0^{(1)},x_0^{(1)}\right) & \Lambda\left(x_0^{(1)},x_0^{(2)}\right)\\ \Lambda\left(x_0^{(1)},x_0^{(2)}\right) & \Lambda\left(x_0^{(2)},x_0^{(2)}\right)\end{bmatrix}.$$

Let $U=[I_2,I_2,\ldots,I_2]^T$ be a $2n\times 2$ matrix with n I_2-blocks. Let $u=\left[\Sigma_0\left(x,x_0^{(1)}\right),\Sigma_0\left(x,x_0^{(2)}\right)\right]^T$ and $v=\left[\Sigma_0\left(x',x_0^{(1)}\right),\Sigma_0\left(x',x_0^{(2)}\right)\right]^T$ be two 2×1 vectors. Then $\Sigma_0\left(\mathcal{X}_n,\mathcal{X}_n\right)=UD_1U^T$, and Γ_n is a block diagonal matrix with n blocks, with each block equal to D_2. Similar to the above argument we have

$$\begin{aligned}\Sigma_n(x,x') &= \Sigma_0(x,x')-\Sigma_0(x,\mathcal{X}_n)\left[S_n\right]^{-1}\Sigma_0(\mathcal{X}_n,x')\\ &= \Sigma_0(x,x')-\Sigma_0(x,\mathcal{X}_n)\left[UD_1U^T+\Gamma_n\right]^{-1}\Sigma_0(\mathcal{X}_n,x')\\ &= \Sigma_0(x,x')-\Sigma_0(x,\mathcal{X}_n)\left[\Gamma_n^{-1}-\Gamma_n^{-1}U\left(D_1^{-1}+U^T\Gamma_n^{-1}U\right)^{-1}U^T\Gamma_n^{-1}\right]\Sigma_0(\mathcal{X}_n,x')\\ &= \Sigma_0(x,x')-n\left[u^TD_2^{-1}v-nu^TD_2^{-1}\left[D_1^{-1}+nD_2^{-1}\right]^{-1}D_2^{-1}v\right]\\ &= \Sigma_0(x,x')-nu^T\left(nD_1+D_2\right)^{-1}v,\end{aligned}$$

where the last line follows from the previous line by the following computation, which uses the matrix identity $A^{-1}B^{-1}=(BA)^{-1}$ and the Sherman-Morrison-Woodbury formula:

$$\begin{aligned}\left[I-nD_2^{-1}(D_1^{-1}+nD_2^{-1})^{-1}\right]D_2^{-1}&=\left[I-n(D_1^{-1}D_2+nI)^{-1}\right]D_2^{-1}=\left[I-\left(I+\frac{D_1^{-1}D_2}{n}\right)^{-1}\right]D_2^{-1}\\&=\left[I-(I-D_1^{-1}(nI+D_2D_1^{-1})^{-1}D_2\right]D_2^{-1}=D_1^{-1}(nI+D_2D_1^{-1})^{-1}=(nD_1+D_2)^{-1}.\end{aligned}$$

Simple algebra then yields $\lim_{n\to+\infty}\ nu^T\left(nD_1+D_2\right)^{-1}v=(d_2+d_3-d_4)/d_1$, where

$$\begin{aligned}d_1&=\Sigma_0\left(x_0^{(1)},x_0^{(1)}\right)\Sigma_0\left(x_0^{(2)},x_0^{(2)}\right)-\left[\Sigma_0\left(x_0^{(1)},x_0^{(2)}\right)\right]^2,\\d_2&=\Sigma_0\left(x_0^{(1)},x_0^{(1)}\right)\Sigma_0\left(x,x_0^{(2)}\right)\Sigma_0\left(x',x_0^{(2)}\right),\\d_3&=\Sigma_0\left(x_0^{(2)},x_0^{(2)}\right)\Sigma_0\left(x,x_0^{(1)}\right)\Sigma_0\left(x',x_0^{(1)}\right),\\d_4&=\Sigma_0\left(x_0^{(1)},x_0^{(2)}\right)\left[\Sigma_0\left(x,x_0^{(1)}\right)\Sigma_0\left(x',x_0^{(2)}\right)+\Sigma_0\left(x,x_0^{(2)}\right)\Sigma_0\left(x',x_0^{(1)}\right)\right].\end{aligned}$$

Under Assumption 2, we always have $d_1>0$ because Σ_0 is positive definite. Specifically, when $x=x'=x_0^{(i)}$ $(i=1,2)$, $[d_2+d_3-d_4]/d_1=\Sigma_0\left(x_0^{(i)},x_0^{(i)}\right)$. Hence $\Sigma_n\left(x_0^{(i)},x_0^{(i)}\right)\to 0$ as $n\to+\infty$ for $i=1,2$. □

LEMMA 7. *If alternative x is sampled infinitely often, then $\Sigma_n(x,x)\to 0$ and $\nu_n^{\mathrm{KG}_\beta}(x)\to 0$ as $n\to\infty$. If alternative $x'\neq x$ is also sampled infinitely often, then $\Sigma_n(x',x')\to 0$ and $\nu_n^{\mathrm{KG}_\beta}(x,x')\to 0$ as $n\to\infty$.*

Proof of Lemma 7. There are k possible decisions in Ξ that involve sampling alternative x, namely, x and (x,x') for $x'\neq x$. Because x is sampled infinitely many times, at least one of these k decisions is chosen infinitely often. Let $\vec{x}$ be one such decision and $\{q_n\}_{n=1}^{\infty}$ be a strictly increasing subsequence of $\mathbb{Z}^+$ such that $\vec{x}_{q_n}=\vec{x}$ for $n=1,2,\ldots$ Because the ordering of the decision-observation pairs can be changed without altering $\Sigma_n(x,x)$, and because taking additional observations can only decrease $\Sigma_n(x,x)$ by Lemma 4, we know that an upper bound on $\Sigma_{q_n}(x,x)$ is given by the posterior variance of θ_x at time n under an allocation rule, call it π, that chooses $x_1=x_2=\cdots=x_n=\vec{x}$. Call this posterior variance $\Sigma_n^{\pi}(x,x)$, so we have $\Sigma_{q_n}(x,x)\leq\Sigma_n^{\pi}(x,x)$. Lemma 6 shows $\lim_{n\to\infty}\Sigma_n^{\pi}(x,x)=0$. Hence $\lim_{n\to\infty}\Sigma_{q_n}(x,x)=0$. Because $\{\Sigma_n(x,x)\}_n$ is a non-negative decreasing sequence, $\lim_{n\to\infty}\Sigma_n(x,x)$ exists and equals 0, due to the uniqueness of the limit. Combining this with Lemma 5 and the non-negativity of the KG_β factors, we have $\lim_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(x)=0$.

If $x'\neq x$ is also sampled infinitely often, similarly we have $\lim_{n\to\infty}\Sigma_n(x',x')=0$, and thus $\lim_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(x,x')=0$ by Lemma 5. □

LEMMA 8. *If $\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(\vec{x})=0$ for all $\vec{x}\in\Xi$, then $\lim_{n\to\infty}\Sigma_n(x,x)=0$ for all x.*

Proof of Lemma 8. Consider an arbitrary sample path on which the μ_n converges to μ_∞. Lemma 2 shows that the set of such sample paths is almost sure. We will show that the claim holds on this sample path.

Lemma 4 shows that $\{\Sigma_n(x,x)\}_n$ is a non-negative decreasing sequence and hence $\lim_{n\to\infty}\Sigma_n(x,x)$ exists and is non-negative for any x. We prove the contrapositive of the statement of the lemma. That is, we suppose that $\max_x\left[\lim_{n\to\infty}\Sigma_n(x,x)\right]>0$ and show that $\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(\vec{x})>0$ for some $\vec{x}\in\Xi$.

We choose two alternatives on which to focus in our analysis. First, because at least one decision $\vec{x}'\in\Xi$ is chosen by the algorithm infinitely often, Lemma 7 shows that there exists an alternative x' with

$\lim_{n\to\infty}\Sigma_n(x',x')=0$. Second, by our choice of sample path, $\lim_{n\to\infty}\mu_n=\mu_\infty$. Let $x^*=\arg\max\mu_\infty$, breaking ties arbitrarily. Then $\mu_\infty(x^*)\geq\mu_\infty(x)$ for all x. It follows that there exists N large enough and a sequence $\{\epsilon_n\}$ decreasing to 0 such that $\mu_n(x^*)\geq\mu_n(x)-\epsilon_n$ for all x for $n\geq N$. If $\lim_{n\to\infty}\Sigma_n(x^*,x^*)>0$, let $x^{(1)}=x^*$ and $x^{(2)}=x'$; otherwise pick $x^{(1)}$ with $\lim_{n\to\infty}\Sigma_n\left(x^{(1)},x^{(1)}\right)>0$ and let $x^{(2)}=x^*$. Let $\vec{x}=\left(x^{(1)},x^{(2)}\right)$.

For each $n\geq N$, let

$$a_n^1=\mu_n\left(x^{(1)}\right),\quad a_n^2=\mu_n\left(x^{(2)}\right),\quad b_n^1=\frac{\Sigma_n\left(x^{(1)},x^{(1)}\right)-\Sigma_n\left(x^{(1)},x^{(2)}\right)}{\sqrt{\beta^{-1}P+Q_n}},\quad b_n^2=\frac{\Sigma_n\left(x^{(2)},x^{(1)}\right)-\Sigma_n\left(x^{(2)},x^{(2)}\right)}{\sqrt{\beta^{-1}P+Q_n}},$$

where P and Q_n are given in (12). Then we have the following:

$$\begin{aligned}
&V_n\left(\vec{x},A_n(\vec{x}),\beta\right)=\mathbb{E}_n\left[\max\mu_{n+1}\left(A_n(\vec{x})\right)\mid\vec{x}_{n+1}=\vec{x},\beta_{n+1}=\beta\right]-\max\mu_n\left(A_n(\vec{x})\right)\\
&\quad\geq\mathbb{E}_n\left[\max\left\{\mu_{n+1}\left(x^{(1)}\right),\mu_{n+1}\left(x^{(2)}\right)\right\}\mid\vec{x}_{n+1}=\vec{x},\beta_{n+1}=\beta\right]-\max\left\{\mu_n\left(x^{(1)}\right),\mu_n\left(x^{(2)}\right)\right\}-\epsilon_n\\
&\quad=\mathbb{E}\left[\max\left\{a_n^1+b_n^1Z,a_n^2+b_n^2Z\right\}\right]-\max\{a_n^1,a_n^2\}-\epsilon_n\\
&\quad=|b_n^1-b_n^2|f\left(-\frac{|a_n^1-a_n^2|}{|b_n^1-b_n^2|}\right)-\epsilon_n,
\end{aligned}\tag{22}$$

where $f(-s)=\varphi(s)-s\Phi(-s)$ is as defined in §3.2, and (22) is understood to be 0 when $|b_n^1-b_n^2|=0$. In this sequence of expressions, the first line applies (13); the second line uses $\max\mu_n(A_n(\vec{x}))\leq\mu_n(x^*)+\epsilon_n=\max\{\mu_n(x^{(1)},\mu_n(x^{(2)})\}+\epsilon_n$ together with the fact that $A_n(\vec{x})$ contains $x^{(1)}$ and $x^{(2)}$; the third line uses (10) and (11); and the last line follows from computations involving the normal distribution, which may be found in equation (14) of Frazier et al. (2009). We will take the limit of (22) as n goes to ∞.

By our choice of sample path, μ_n converges to μ_∞, so $\lim_{n\to\infty}|a_n^1-a_n^2|=|\mu_\infty(x^{(1)})-\mu_\infty(x^{(2)})|:=\gamma_1<\infty$. We now show that $\lim_{n\to\infty}|b_n^1-b_n^2|$ is strictly positive. First,

$$|b_n^1-b_n^2|=\frac{|\Sigma_n\left(x^{(1)},x^{(1)}\right)-2\Sigma_n\left(x^{(1)},x^{(2)}\right)+\Sigma_n\left(x^{(2)},x^{(2)}\right)|}{\sqrt{\beta^{-1}P+Q_n}}=\frac{|Q_n|}{\sqrt{\beta^{-1}P+Q_n}}.$$

Then, $\left\{\Sigma_n\left(x^{(1)},x^{(1)}\right)\right\}_n$ is bounded above by $\Sigma_0(x^{(1)},x^{(1)})$ by Lemma 4, and $\lim_{n\to\infty}\Sigma_n\left(x^{(2)},x^{(2)}\right)=0$, so $\lim_{n\to\infty}|\Sigma_n\left(x^{(1)},x^{(2)}\right)|\leq\lim_{n\to\infty}\sqrt{\Sigma_n\left(x^{(1)},x^{(1)}\right)\Sigma_n\left(x^{(2)},x^{(2)}\right)}=0$. Hence $\lim_{n\to\infty}\Sigma_n\left(x^{(1)},x^{(2)}\right)=0$. It follows that $\lim_{n\to\infty}Q_n=\lim_{n\to\infty}\Sigma_n\left(x^{(1)},x^{(1)}\right)-2\Sigma_n\left(x^{(1)},x^{(2)}\right)+\Sigma_n\left(x^{(2)},x^{(2)}\right)=\lim_{n\to\infty}\Sigma_n(x^{(1)},x^{(1)})$, which is strictly positive by the construction of $x^{(1)}$. Thus,

$$\lim_{n\to\infty}|b_n^1-b_n^2|=\liminf_{n\to\infty}\frac{|Q_n|}{\sqrt{\beta^{-1}P+Q_n}}=\frac{|\lim_{n\to\infty}\Sigma_n(x^{(1)},x^{(1)})|}{\sqrt{\beta^{-1}P+\lim_{n\to\infty}\Sigma_n(x^{(1)},x^{(1)})}}:=\gamma_2>0.$$

Recall (22). The function $s\mapsto f(-s)$ is continuous, so (22) is continuous in $(|a_n^1-a_n^2|,|b_n^1-b_n^2|)$ on $[0,\infty)\times(0,\infty)$, and the limit of (22) as $n\to\infty$ is $\gamma_2 f\left(-\gamma_1/\gamma_2\right)$. Since $V_n(\vec{x},A_n(\vec{x}),\beta)$ is bounded below by (22),

$$\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}\left(\vec{x}\right)=\frac{\liminf_{n\to\infty}V_n\left(\vec{x},A_n(\vec{x}),\beta\right)}{\beta c\left(\vec{x}\right)}\geq\frac{1}{\beta c\left(\vec{x}\right)}\gamma_2 f\left(-\frac{\gamma_1}{\gamma_2}\right)>0,$$

where we have used that $\gamma_1<\infty$ and $\gamma_2>0$, and $f(-s)$ is strictly positive for $s<\infty$. □

Proof of Theorem 1 We first show, by contradiction, that $\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}\left(\vec{x}\right)=0$ almost surely for all $\vec{x}\in\Xi$. Consider an arbitrary sample path of the KG_β^2 algorithm from the almost sure set on which the claim of Lemma 8 holds. Let

$$\chi_0:=\left\{\vec{x}\in\Xi:\lim_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}\left(\vec{x}\right)\text{ exists and is }0\right\}\qquad\text{and}\qquad\chi_1:=\left\{\vec{x}\in\Xi:\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}\left(\vec{x}\right)=0\right\},$$

Suppose for contradiction that $\chi_1 \neq \Xi$, i.e., that $\Xi \setminus \chi_1 = \left\{\vec{x} \in \Xi : \liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) > 0\right\}$ is not empty.

Pick $\vec{x} \in \Xi \setminus \chi_1$. By Condition 1, there exists a subsequence of $\mathbb{Z}^+$, denoted by $\{n_i\}_{i=1}^\infty$, such that $\vec{x}_{n_i} \in \Xi_{n_i}$ for all i. Also, $\liminf_{i\to\infty} \nu_{n_i}^{\mathrm{KG}_\beta}(\vec{x}) \geq \liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) > 0$. Thus there exists some $\epsilon > 0$ and a subsequence of $\{n_i\}_{i=1}^\infty$, denoted $\{n_j\}_{j=1}^\infty$, such that $\nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}) \geq \epsilon$ for all j. Then $\nu_{n_j}^{\mathrm{KG}_\beta}\left(\vec{x}_{n_j}\right) \geq \nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}) \geq \epsilon$ for all j.

For each $\vec{x}' \in \Xi \setminus \chi_0$, the contrapositive of Lemma 7 implies there exists a finite number $N(\vec{x}')$ such that the KG_β^2 algorithm does not choose $\vec{x}'$ for $n > N(\vec{x}')$. Let $N := \max_{\vec{x}' \in \Xi\setminus\chi_0} N(\vec{x}')$. Then $\vec{x}_n \in \chi_0$ for all $n > N$.

For each $\vec{x}' \in \chi_0$, $\lim_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}') = 0$. Hence there exists a finite number $N_0(\vec{x}')$ such that $\nu_n^{\mathrm{KG}_\beta}(\vec{x}') < \epsilon$ for all $n > N_0(\vec{x}')$. Let $N_0 := \max_{\vec{x}' \in \chi_0} N_0(\vec{x}')$. Then for all $n > N_0$, $\nu_n^{\mathrm{KG}_\beta}(\vec{x}') < \epsilon$ for any $\vec{x}' \in \chi_0$.

It follows that $\nu_n^{\mathrm{KG}_\beta}(\vec{x}_n) < \epsilon$ for all $n > \max\{N_0, N\}$, which contradicts $\nu_{n_j}^{\mathrm{KG}_\beta}\left(\vec{x}_{n_j}\right) \geq \epsilon$ for all j. We thus conclude that $\chi_1 = \Xi$ on this sample path, i.e. $\liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) = 0$ for all $\vec{x} \in \Xi$. Since the chosen sample path was arbtitrary, this holds almost surely.

Since we chose a sample path on which Lemma 8 holds, $\lim_{n\to\infty} \Sigma_n(x,x) = 0$ on this sample path. Moreover, as the set of sample paths on which Lemma 8 holds is almost sure, $\lim_{n\to\infty} \Sigma_n(x,x) = 0$ almost surely.

To show that $\lim_n \mu_n(x) = \theta(x)$ almost surely for each x, we first show this limit holds in L^2. For each x, $E\left[(\mu_n(x) - \theta(x))^2\right] = E\left[E_n\left[(\mu_n(x) - \theta(x))^2\right]\right] = E\left[\Sigma_n(x,x)\right]$. Taking the limit as $n \to \infty$ and using $0 \leq \Sigma_n(x,x) \leq \Sigma_0(x,x)$ with the dominated convergence theorem implies $\lim_{n\to\infty} E\left[(\mu_n(x) - \theta(x))^2\right] = E\left[\lim_{n\to\infty} \Sigma_n(x,x)\right] = 0$. Then, since $\mu_n(x)$ converges to $\theta(x)$ in L^2, and Lemma 2 implies $\lim_{n\to\infty} \mu_n(x)$ exists almost surely, this almost sure limit equals $\theta(x)$.

We now show that $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely. First, $x^* \in \arg\max_x \theta(x)$ is almost surely unique as a realization of a multivariate normal random variable, and so $\epsilon = \theta(x^*) - \max_{x \neq x^*} \theta(x)$ is almost surely strictly positive. Fix a sample path on which $\lim_{n\to\infty} \mu_n(x) = \theta(x)$ for each x (which occurs almost surely). There exists $N < \infty$ such that $|\mu_n - \theta(x)| < \frac{\epsilon}{2}$ for all $n > N$. Then, for all $n > N$ and all $x \neq x^*$, $\mu_n(x^*) > \theta(x^*) - \frac{\epsilon}{2} > \theta(x) + \frac{\epsilon}{2} > \mu_n(x)$, implying x^* is the unique element in $\arg\max_x \mu_n(x)$. This shows that $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely. □

B. MLE for Unknown Parameters in §6.2

This section derives the MLE used in §6.2to estimate the parameters η, σ_0^2, $\vec{\alpha} = \{\alpha_i\}_{1\leq i\leq d}$, σ_ϵ^2 and ρ, which determine Λ, μ_0 and Σ_0 through the model defined in §6.1. The derivation is related to results of Huang et al. (2006) and Rasmussen and Williams (2006, Sections 2 and 5), but it goes beyond this previous work in considering the sampling correlation ρ.

Continue to let $\mathcal{X}_n$ and $\mathcal{Y}_n$ represent all design points and outputs that have been observed through stage n, including the initialization phase. Let $m = |\mathcal{X}_n|$ be the total number of observations. Set $g = \sigma_0^2/\sigma^2$, $\sigma^2 = \sigma_0^2 + \sigma_\epsilon^2$, and let δ_{ij} be 1 if $\mathcal{X}_n^{(i)}$ and $\mathcal{X}_n^{(j)}$ are sampled using CRN, and 0 otherwise. Then $\mathcal{Y}_n \sim \mathcal{N}\left(\eta\vec{1}, \sigma^2 R\right)$ for a correlation matrix R defined by

$$R(i,j) = \begin{cases} 1, & \text{if } i = j, \\ g\exp\left\{-\sum_{l=1}^d \alpha_l \left[\zeta_l\left(\mathcal{X}_n^{(i)}\right) - \zeta_l\left(\mathcal{X}_n^{(j)}\right)\right]^2\right\} + (1-g)\rho\delta_{ij}, & \text{if } i \neq j. \end{cases}$$

The MLE is then $\arg\max_{\eta,\sigma_0^2,\vec{\alpha},\sigma_\epsilon^2,\rho} \log p\left(\mathcal{Y}_n \mid \eta, \sigma_0^2, \vec{\alpha}, \sigma_\epsilon^2, \rho\right)$. We reparameterize this problem by replacing $(\sigma_0^2, \sigma_\epsilon^2)$ with (g, σ^2), which uniquely determine each other, to obtain an equivalent formulation of the MLE,

$$\arg\max_{\eta,\sigma^2,g,\vec{\alpha},\rho} \log p\left(\mathcal{Y}_n \mid \eta, \sigma_0^2, \vec{\alpha}, \sigma_\epsilon^2, \rho\right) = \arg\max_{\eta,\sigma^2,g,\vec{\alpha},\rho} \log p\left(\mathcal{Y}_n \mid \eta, \sigma_0^2, R\right),$$

where we have noted that the parameters $g, \vec{\alpha}, \rho$ only influence the log-likelihood through the correlation matrix R, which is determined by them.

We solve this optimization problem in two steps, first optimizing over σ^2 and η with the other parameters fixed, which can be done analytically, and then numerically optimizing the resulting value over the set of R matrices that can be achieved with the remaining parameters g, $\vec{\alpha}$,ρ. We first describe optimization over σ^2 and η in the following lemma.

LEMMA 9. *The maximum log-likelihood over η and σ^2 with R fixed is*

$$\log p\left(\mathcal{Y}_n \mid \widehat{\eta}, \widehat{\sigma}^2, R\right) = \max_{\eta,\sigma^2} \log p\left(\mathcal{Y}_n \mid \eta, \sigma^2, R\right) = -\frac{1}{2}\left(m \log \widehat{\sigma}^2 + \log|R|\right) - \frac{m}{2}(1 + \log 2\pi),$$

where $\vec{1}$ denotes a length-m column vector of ones, $|R|$ is the determinant of R, and

$$\widehat{\sigma}^2 = \frac{1}{n}\left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)^T R^{-1}\left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right) \qquad \widehat{\eta} = \left(\vec{1}^T R^{-1}\vec{1}\right)^{-1}\vec{1}^T R^{-1}\mathcal{Y}_n. \tag{23}$$

Proof of Lemma 9. We first rewrite the log-likelihood as

$$\begin{aligned}\log p\left(\mathcal{Y}_n \mid \eta, \sigma^2, R\right) &= -\frac{1}{2}\left(\mathcal{Y}_n - \eta\vec{1}\right)^T \left(\sigma^2 R\right)^{-1}\left(\mathcal{Y}_n - \eta\vec{1}\right) - \frac{1}{2}\log|\sigma^2 R| - \frac{m}{2}\log 2\pi \\ &= -\frac{1}{2\sigma^2}\left(\mathcal{Y}_n - \eta\vec{1}\right)^T R^{-1}\left(\mathcal{Y}_n - \eta\vec{1}\right) - \frac{m}{2}\log\sigma^2 - \frac{1}{2}\log|R| - \frac{m}{2}\log 2\pi\end{aligned}$$

Observe that $\widehat{\eta} = \arg\min_\eta \left[\left(\mathcal{Y}_n - \eta\vec{1}\right)^T R^{-1}\left(\mathcal{Y}_n - \eta\vec{1}\right)\right] = \left(\vec{1}^T R^{-1}\vec{1}\right)^{-1}\vec{1}^T R^{-1}\mathcal{Y}_n$ is the generalized least squares estimate of η. Let $C := \left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)^T R^{-1}\left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)$. We then consider a function $H : \mathbb{R}^+ \mapsto \mathbb{R}$ with $H(s) = C/s + m\log s$. Since $H'(s) = -C/s^2 + m/s$, we know C/m is the global minimum of H. It follows that $\widehat{\sigma}^2 = C/m$ is the MLE of σ^2. We thus conclude that $\log p\left(\mathcal{Y}_n \mid \widehat{\eta}, \widehat{\sigma}^2, R\right) = -\frac{1}{2}\left(m\log\widehat{\sigma}^2 + \log|R|\right) - \frac{m}{2}(1 + \log 2\pi)$ is the maximum log marginal likelihood of $\mathcal{Y}_n$ given matrix R. □

To complete the calculation of the MLE, we maximize the expression for $\log p\left(\mathcal{Y}_n \mid \widehat{\eta}, \widehat{\sigma}^2, R\right)$ from Lemma 9 over matrices R that can be obtained by varying the remaining parameters g, ρ and $\vec{\alpha}$. Denote such maximizers by $\widehat{g}$, $\widehat{\rho}$ and $\widehat{\vec{\alpha}}$. To find them, we examine the partial derivatives of $\log p\left(\mathcal{Y}_n \mid \widehat{\eta}, \widehat{\sigma}^2, R\right)$ with respect to g, ρ and α_l $(l = 1, \ldots, d)$. Let t denote any of these parameters. Rasmussen and Williams (2006, Sec. 5) show $\frac{\partial R^{-1}}{\partial t} = -R^{-1}\frac{\partial R}{\partial t}R^{-1}$ and $\frac{\partial \log|R|}{\partial t} = \operatorname{tr}\left(R^{-1}\frac{\partial R}{\partial t}\right)$. Thus, we can write

$$\frac{\partial}{\partial t}\log p\left(\mathcal{Y}_n \mid R\right) = -\frac{1}{2}\left[\left(\frac{n}{\widehat{\sigma}^2}\frac{\partial\widehat{\sigma}^2}{\partial t}\right) + \operatorname{tr}\left(R^{-1}\frac{\partial R}{\partial t}\right)\right],$$

where

$$\begin{aligned}\frac{\partial\widehat{\sigma}^2}{\partial t} &= -\frac{1}{n}\left(2\frac{\partial\widehat{\eta}}{\partial t}\left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)^T R^{-1}\vec{1} + \left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)^T R^{-1}\frac{\partial R}{\partial t}R^{-1}\left(\mathcal{Y}_n - \widehat{\eta}\vec{1}\right)\right), \\ \frac{\partial\widehat{\eta}}{\partial t} &= \left(\vec{1}^T R^{-1}\vec{1}\right)^{-2}\vec{1}^T R^{-1}\frac{\partial R}{\partial t}R^{-1}\vec{1}\vec{1}^T R^{-1}\mathcal{Y}_n - \left(\vec{1}^T R^{-1}\vec{1}\right)^{-1}\vec{1}^T R^{-1}\frac{\partial R}{\partial t}R^{-1}\mathcal{Y}_n.\end{aligned}$$

Each entry of the matrix $\frac{\partial R}{\partial t}$ is given by $\frac{\partial R}{\partial t}(i,i) = 0$ for $i = 1, 2, \ldots, m$ and, for $i \neq j$,

$$\begin{aligned}\frac{\partial R}{\partial g}(i,j) &= \exp\left\{-\sum_{l=1}^d \alpha_l\left[\zeta_l\left(\mathcal{X}_n^{(i)}\right) - \zeta_l\left(\mathcal{X}_n^{(j)}\right)\right]^2\right\} - \rho\delta_{ij}, \\ \frac{\partial R}{\partial \alpha_l}(i,j) &= -g\left[\zeta_l\left(\mathcal{X}_n^{(i)}\right) - \zeta_l\left(\mathcal{X}_n^{(j)}\right)\right]^2 \exp\left\{-\sum_{l=1}^d \alpha_l\left[\zeta_l\left(\mathcal{X}_n^{(i)}\right) - \zeta_l\left(\mathcal{X}_n^{(j)}\right)\right]^2\right\}, \quad l = 1, 2, \ldots, d, \\ \frac{\partial R}{\partial \rho}(i,j) &= (1-g)\delta_{ij}.\end{aligned}$$

By applying a Cholesky decomposition to the positive definite matrix R, one can avoid a direct inversion of R in the computations above by solving triangular linear systems. Letting G be the Cholesky factor, the log determinant of R can be calculated efficiently by $\log|R| = 2\sum_{1=1}^{m} \log G_{ii}$.

With these expressions, we can then use gradient based maximization methods to find $\widehat{g}$, $\widehat{\rho}$ and $\widehat{\vec{\alpha}}$. As previously discussed, MLEs $\widehat{\eta}$ and $\widehat{\sigma}^2$ are given by (23), and MLEs $\widehat{\sigma}_0^2$, $\widehat{\sigma}_\epsilon^2$ follow from inverting the definitions of g and σ^2 and applying the inverted expressions to $\widehat{g}$ and $\widehat{\sigma}^2$.

C. Gradients Results

This section provides details to support the computation of gradients of the posterior means and predictive covariances with respect to sampling decisions (singletons or pairs), under the assumption that the alternatives sampled are embedded in $\mathbb{R}^d$. These were used in §6.3 to compute the gradient of the VOI and KG factor with respect to the location of the sampling decision. We also demonstrate simplifications of those results for the special case of a GP prior distribution with Gaussian kernel and constant mean, and a sampling covariance that satisfies a compound sphericity assumption (as in §6.1 and §6.2).

We continue the notational convention of §6.3, in which derivatives taken with respect to x (in the case of singletons) and $\vec{x}$ (in the case of pairs), actually indicate derivatives taken with respect to these alternatives' grid coordinates: $(\zeta_i(x) : i = 1, \ldots, d)$ for singletons; and $(\zeta_i(x^{(1)}), \zeta_i(x^{(2)}) : i = 1, \ldots, d)$ for pairs.

The expressions provided in C.1 and C.2 are for general priors and sampling models, and have within them terms such as $\nabla_x[\mu_0(x)]$, $\nabla_x[\Sigma_0(x,x)]$, and $\nabla_x[\Lambda(x,x)]$, whose values depend on the specific prior and form of sampling correlation assumed. Specific values for these quantities for the prior and sampling correlation used in §6.2-§6.3 are provided in Appendix C.3.

C.1. Gradients of $\mu_n(x')$ and $\widetilde{\sigma}_n(x,x',\beta)$ when Sampling a Singleton

In this section, we provide expressions for $\nabla_x[\mu_n(x')]$ and $\nabla_x[\widetilde{\sigma}_n(x,x',\beta)]$ for an arbitrary alternative x'. These expressions can be substituted in (19) to obtain an expression for the gradient of the VOI $V_n(x, A_n(x), \beta)$ when sampling a singleton $\vec{x} = x$, that holds when $A_n(x)$ is as described in §6.3.

To support this computation, let $J_n(x') := \nabla_{x'}[\Sigma_0(x', \mathcal{X}_n)]$ be a $d \times |\mathcal{X}_n|$ matrix, the ith column of which is $\nabla_{x'}[\Sigma_0(x', \mathcal{X}_n(i))]$, where $\mathcal{X}_n(i)$ is the ith entry of $\mathcal{X}_n$. Recall $\widetilde{\mathcal{Y}}_n$, S_n and $K_n(\vec{x})$ from (5).

We first provide an expression for $\nabla_x[\mu_n(x')]$.

LEMMA 10. *$\nabla_x[\mu_n(x')] = \nabla_x[\mu_0(x)] + J_n(x)[S_n]^{-1}\widetilde{\mathcal{Y}}_n$ if $x = x'$, and is $\vec{0}$ if $x \neq x'$.*

Proof of Lemma 10. If $x \neq x'$, then $\mu_n(x')$ does not depend on x, so $\nabla_x[\mu_n(x')] = 0$. Now consider $x = x'$. Note that x is the last element of $\mathcal{X}_{n,x}$. Let e_x be a column vector $[0, 0, \ldots, 0, 1]^T$ with length $|\mathcal{X}_n| + 1$. Then

$$\begin{aligned}\mu_n(x) &= e_x^T \mu_n(\mathcal{X}_{n,x}) = e_x^T \left[\mu_0(\mathcal{X}_{n,x}) + K_n(x)\widetilde{\mathcal{Y}}_n\right] \\ &= e_x^T \mu_0(\mathcal{X}_{n,x}) + e_x^T \Sigma_0(\mathcal{X}_{n,x}, \mathcal{X}_{n,x}) \left[I_{|\mathcal{X}_n|}, \vec{0}\right]^T [S_n]^{-1}\widetilde{\mathcal{Y}}_n = \mu_0(x) + \Sigma_0(x, \mathcal{X}_n)[S_n]^{-1}\widetilde{\mathcal{Y}}_n,\end{aligned}$$

where we use (5) and (6) in the last line. Because $[S_n]^{-1}\widetilde{\mathcal{Y}}_n$ does not depend on x, the gradient is $\nabla_x[\mu_n(x)] = \nabla_x(\mu_{0x}) + \nabla_x[\Sigma_0(x, \mathcal{X}_n)][S_n]^{-1}\widetilde{\mathcal{Y}}_n = \nabla_x[\mu_0(x)] + J_n(x)[S_n]^{-1}\widetilde{\mathcal{Y}}_n$. □

We now provide an expression for $\nabla_x[\widetilde{\sigma}_n(x,x',\beta)]$.

LEMMA 11.

$$\nabla_x\left[\widetilde{\sigma}_n\left(x,x',\beta\right)\right]=\frac{B\nabla_x\left[\Sigma_n(x',x)\right]-\Sigma_n\left(x',x\right)\nabla_x(B)}{B^2}$$

where $B:=\sqrt{\Lambda(x,x)/\beta+\Sigma_n(x,x)}$, *and*

$$\nabla_x\left[\Sigma_n\left(x',x\right)\right]=\begin{cases}\nabla_x\left[\Sigma_0(x',x)\right]-J_n(x)[S_n]^{-1}\Sigma_0(\mathcal{X}_n,x'), & \text{if } x'\neq x,\\ \nabla_x\left[\Sigma_0(x,x)\right]-2J_n(x)[S_n]^{-1}\Sigma_0(\mathcal{X}_n,x), & \text{if } x'=x,\end{cases}$$

$$\nabla_x(B)=\frac{1}{2B}\left\{\nabla_x\left[\Lambda(x,x)/\beta+\Sigma_0(x,x)\right]-2J_n(x)[S_n]^{-1}\Sigma_0(\mathcal{X}_n,x)\right\}.$$

Proof of Lemma 11. Recall that $\widetilde{\sigma}_{nx'}(X,\beta)=\Sigma_n(x',x)/B$, so $\nabla_x\left[\widetilde{\sigma}_n\left(x,x',\beta\right)\right]$ is as claimed. Next, recall from Lemma 3 that $\Sigma_n(x',x)=\Sigma_0(x',x)-\Sigma_0\left(x,\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right)$. Thus if $x'\neq x$, then

$$\nabla_x\left[\Sigma_n(x',x)\right]=\nabla_x\left[\Sigma_0(x',x)\right]-\nabla_x\left[\Sigma_0(x,\mathcal{X}_n)\right][S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right)=\nabla_x\left[\Sigma_0(x',x)\right]-J_n(x)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right).$$

If $x'=x$, then using standard matrix differentiation, we can compute the gradient as $\nabla_x\left[\Sigma_n(x,x)\right]=\nabla_x\left[\Sigma_0(x,x)\right]-2J_n(x)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x\right)$. The claimed formula for $\nabla_x(B)$ follows from simple algebra. □

C.2. Gradients of $\mu_n\left(x'\right)$ and $\widetilde{\sigma}_n\left(\vec{x},x',\beta\right)$ when Sampling a Pair

In this section, we describe computation of $\nabla_{\vec{x}}\left[\mu_n\left(x'\right)\right]$ and $\nabla_{\vec{x}}\left[\widetilde{\sigma}_n\left(\vec{x},x',\beta\right)\right]$ for an arbitrary alternative x'. These expressions can be substituted in (20) to obtain an expression for the gradient of the VOI $V_n\left(x,A_n(x),\beta\right)$ when sampling a pair $\vec{x}$, that holds when $A_n(x)$ is as described in §6.3.

The gradient $\nabla_{x^{(i)}}\left[\mu_n(x')\right]$ for $i=1,2$ is given in Lemma 10 where we replace x by $x^{(i)}$. The derivation is similar and is hence omitted. The derivation of $\nabla_{x^{(i)}}\left[\widetilde{\sigma}_n\left(\vec{x},x',\beta\right)\right]$ when sampling pairs differs from that of the gradient when sampling a singleton, so details follow.

LEMMA 12. *For* $i=1,2$,

$$\begin{aligned}\nabla_{x^{(i)}}\left[\widetilde{\sigma}_n\left(\vec{x},x',\beta\right)\right]=\frac{1}{B^2}\Big\{&B\nabla_{x^{(i)}}\left[\Sigma_n\left(x',x^{(1)}\right)-\Sigma_n\left(x',x^{(2)}\right)\right]\\&-\left[\Sigma_n\left(x',x^{(1)}\right)-\Sigma_n\left(x',x^{(2)}\right)\right]\nabla_{x^{(i)}}(B)\Big\},\end{aligned}\tag{24}$$

where

$$\begin{aligned}B:=\Big\{&\beta^{-1}\left[\Lambda\left(x^{(1)},x^{(1)}\right)+\Lambda\left(x^{(2)},x^{(2)}\right)-2\Lambda\left(x^{(1)},x^{(2)}\right)\right]\\&+\Sigma_n\left(x^{(1)},x^{(1)}\right)+\Sigma_n\left(x^{(2)},x^{(2)}\right)-2\Sigma_n\left(x^{(1)},x^{(2)}\right)\Big\}^{\frac{1}{2}},\end{aligned}\tag{25}$$

$$\begin{aligned}&\nabla_{x^{(i)}}\left[\Sigma_n\left(x',x^{(1)}\right)-\Sigma_n\left(x',x^{(2)}\right)\right]\\&=\begin{cases}\nabla_{x^{(1)}}\left[\Sigma_0\left(x',x^{(1)}\right)\right]-J_n\left(x^{(1)}\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right), & \text{if } i=1,x'\neq x^{(1)}\\ \nabla_{x^{(1)}}\left[\Sigma_0\left(x^{(1)},x^{(1)}\right)-\Sigma_0\left(x^{(1)},x^{(2)}\right)\right]-J_n\left(x^{(1)}\right)[S_n]^{-1}\left[2\Sigma_0\left(\mathcal{X}_n,x^{(1)}\right)-\Sigma_0\left(\mathcal{X}_n,x^{(2)}\right)\right], & \text{if } i=1,x'=x^{(1)}\\ -\nabla_{x^{(2)}}\left[\Sigma_0\left(x',x^{(2)}\right)\right]+J_n\left(x^{(2)}\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right), & \text{if } i=2,x'\neq x^{(2)}\\ \nabla_{x^{(2)}}\left[\Sigma_0\left(x^{(1)},x^{(2)}\right)-\Sigma_0\left(x^{(2)},x^{(2)}\right)\right]+J_n\left(x^{(2)}\right)[S_n]^{-1}\left[2\Sigma_0\left(\mathcal{X}_n,x^{(2)}\right)-\Sigma_0\left(\mathcal{X}_n,x^{(1)}\right)\right], & \text{if } i=2,x'=x^{(2)}\end{cases}\end{aligned}\tag{26}$$

and

$$\begin{aligned}\nabla_{x^{(i)}}(B)=\frac{1}{B}\Bigg\{&\nabla_{x^{(i)}}\left[\frac{1}{2}\left[\beta^{-1}\Lambda\left(x^{(i)},x^{(i)}\right)+\Sigma_0\left(x^{(i)},x^{(i)}\right)\right]-\left[\beta^{-1}\Lambda\left(x^{(1)},x^{(2)}\right)+\Sigma_0\left(x^{(1)},x^{(2)}\right)\right]\right]\\&+J_n\left(x^{(i)}\right)[S_n]^{-1}\left[\Sigma_0\left(\mathcal{X}_n,x^{3-i}\right)-\Sigma_0\left(\mathcal{X}_n,x^{i}\right)\right]\Bigg\}.\end{aligned}$$

Proof of Lemma 12. First, recall that $\widetilde{\sigma}_n\left(\vec{x},x',\beta\right)=\frac{1}{B}\left[\Sigma_n\left(x',x^{(1)}\right)-\Sigma_n\left(x',x^{(2)}\right)\right]$, hence (24) follows. Using (5) and (6), similar to the proof of Lemma 11, we have for $i=1,2$ that $\Sigma_n\left(x',x^{(i)}\right)=\Sigma_0\left(x',x^{(i)}\right)-\Sigma_0\left(x^{(i)},\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right)$.

We show the first two cases ($i=1$) of (26). The other two cases ($i=2$) follow similarly, and are ommitted. In the first case ($x'\neq x^{(1)}$) we have $\nabla_{x^{(1)}}\left[\Sigma_n\left(x',x^{(1)}\right)-\Sigma_n\left(x',x^{(2)}\right)\right]=\nabla_{x^{(1)}}\left[\Sigma_n\left(x',x^{(1)}\right)\right]=\nabla_{x^{(1)}}\left[\Sigma_0\left(x',x^{(1)}\right)\right]-J_n\left(x^{(1)}\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x'\right)$. In the second case ($x'=x^{(1)}$) then from the observation that $\Sigma_n\left(x^{(1)},x^{(1)}\right)-\Sigma_n\left(x^{(1)},x^{(2)}\right)=\Sigma_0\left(x^{(1)},x^{(1)}\right)-\Sigma_0\left(x^{(1)},x^{(2)}\right)-\Sigma_0\left(x^{(1)},\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x^{(1)}\right)+\Sigma_0\left(x^{(1)},\mathcal{X}_n\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x^{(2)}\right)$, it follows from standard matrix differentiation and the definition of $J_n\left(x^{(1)}\right)$ that $\nabla_{x^{(1)}}\left[\Sigma_n\left(x^{(1)},x^{(1)}\right)-\Sigma_n\left(x^{(1)},x^{(2)}\right)\right]=\nabla_{x^{(1)}}\left[\Sigma_0\left(x^{(1)},x^{(1)}\right)-\Sigma_0\left(x^{(1)},x^{(2)}\right)\right]-2J_n\left(x^{(1)}\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x^{(1)}\right)+J_n\left(x^{(1)}\right)[S_n]^{-1}\Sigma_0\left(\mathcal{X}_n,x^{(2)}\right)$.

It remains to compute $\nabla_{x^{(i)}}(B)$. Notice that for $i=1$,

$$\begin{aligned}
&\nabla_{x^{(1)}}\left[\Sigma_n\left(x^{(1)},x^{(1)}\right)+\Sigma_n\left(x^{(2)},x^{(2)}\right)-2\Sigma_n\left(x^{(1)},x^{(2)}\right)\right]\\
&\quad=\nabla_{x^{(1)}}\left[\Sigma_n\left(x^{(1)},x^{(1)}\right)-\Sigma_n\left(x^{(1)},x^{(2)}\right)\right]-\nabla_{x^{(1)}}\left[\Sigma_n\left(x^{(2)},x^{(1)}\right)-\Sigma_n\left(x^{(2)},x^{(2)}\right)\right]\\
&\quad=\nabla_{x^{(1)}}\left[\Sigma_0\left(x^{(1)},x^{(1)}\right)-2\Sigma_0\left(x^{(1)},x^{(2)}\right)\right]+2J_n\left(x^{(1)}\right)[S_n]^{-1}\left[\Sigma_0\left(\mathcal{X}_n,x^{(2)}\right)-\Sigma_0\left(\mathcal{X}_n,x^{(1)}\right)\right],
\end{aligned}$$

where the last equation follows from (26). $\nabla_{x^{(1)}}(B)$ then follows from the definition of B. The formula for $\nabla_{x^{(2)}}(B)$ is similar. □

C.3. Simplification under Compound Sphericity, Constant Prior Mean, and Gaussian Kernel

The gradients of the VOI and KG factors in Appendices C.1-C.2 involve the gradients of the sampling covariance matrix, and of the mean and covariance for the unknown mean θ. That is, they include the terms $\nabla_x[\Lambda(x,x')]$, $\nabla_x[\mu_0(x)]$ and $\nabla_x[\Sigma_0(x,x')]$ for arbitrary x,x'. These values depend on the prior distribution and the assumed form of the sampling correlation.

In this section, we provide specific values for these quantities that result from adopting the modeling choices from §6.2-§6.3: a GP prior with a Gaussian kernel and constant mean, and compound sphericity. These choices substantially simplify the expressions from Appendices C.1-C.2, as many terms become 0.

First, under compound sphericity, $\nabla_x[\Lambda(x,x')]=\vec{0}$ for arbitrary x and x'. Second, under constant prior mean, $\nabla_x[\mu_0(x)]=\nabla_x[\eta]=\vec{0}$. Third, we compute $\nabla_x[\Sigma_0(x,x')]$ for arbitrary x and x'. Denote by $\circ$ the Hadamard (componentwise) product of two vectors u and v of the same length, so that $(u\circ v)(i)=u(i)v(i)$. Then $\nabla_x[\Sigma_0(x,x')]=2\Sigma_0(x,x')\,\alpha\circ[\zeta(x)-\zeta(x')]$. In particular, $\nabla_x[\Sigma_0(x,x)]=\nabla_x[\sigma_0^2]=\vec{0}$.

Acknowledgments

We thank Jeff Hong and Barry Nelson for discussions about the assemble to order model and ISC. Peter Frazier was supported by AFOSR FA9550-11-1-0083 and FA9550-12-1-0200, and by NSF IIS-1247696, IIS-142251 and CMMI-1254298. Jing Xie was supported by AFOSR FA9550-11-1-0083.

References

Andradóttir, S. 1998. Simulation optimization. *Handbook of simulation: Principles, methodology, advances, applications, and practice*. Wiley-Interscience, New York, 307–333.

Andradóttir, S. 2006. An overview of simulation optimization via random search. S.G. Henderson, B.L. Nelson, eds., *Handbooks in Operations Research and Management Science: Simulation*. North-Holland, 617–631.

Ankenman, B., B.L. Nelson, J. Staum. 2010. Stochastic Kriging for Simulation Metamodeling. *Operations Research* **58**(2) 371–382.

Barton, R.R. 2009. Simulation optimization using metamodels. M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, R.G. Ingalls, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 230–238.

Branke, J., S.E. Chick, C. Schmidt. 2007. Selecting a selection procedure. *Management Science* **53**(12) 1916–1932.

Brochu, E., V.M. Cora, N. de Freitas. 2009. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Tech. Rep. TR-2009-23, Department of Computer Science, University of British Columbia.

Chen, C.H., L.H. Lee. 2010. *Stochastic simulation optimization: an optimal computing budget allocation*. World Scientific.

Chen, X., B.E. Ankenman, B.L. Nelson. 2012. The effects of common random numbers on stochastic kriging metamodels. *ACM TOMACS* **22**(7) 1–20.

Chen, X., B.E. Ankenman, B.L. Nelson. 2013. Enhancing stochastic kriging metamodels with gradient estimators. *Operations Reesarch* **61** in press.

Chick, S.E. 2006. Bayesian ideas and discrete event simulation: why, what and how. L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 96–105.

Chick, S.E., P.I. Frazier. 2012. Sequential sampling for selection with economics of selection procedures. *Management Science* **58**(3) 550–569.

Chick, S.E., K. Inoue. 2001a. New procedures to select the best simulated system using common random numbers. *Management Science* **47**(8) 1133–1149.

Chick, S.E., K. Inoue. 2001b. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research* **49**(5) 732–743.

Clark, G.M., W. Yang. 1986. A Bonferroni selection procedure when using common random numbers with unknown variances. J. Wilson, J. Henriksen, S. Roberts, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 313–315.

Cressie, N.A.C. 1993. *Statistics for Spatial Data, revised edition*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley Interscience, New York.

Forrester, A., A. Sobester, A. Keane. 2008. *Engineering design via surrogate modelling: a practical guide*. Wiley, West Sussex, UK.

Frazier, P. 2009–2010. http://people.orie.cornell.edu/pfrazier/src.html.

Frazier, P., W. B. Powell, S. Dayanik. 2009. The knowledge gradient policy for correlated normal beliefs. *INFORMS Journal on Computing* **21**(4) 599–613.

Frazier, P.I. 2010. Decision-theoretic foundations of simulation optimization. *Wiley Encyclopedia of Operations Research and Management Science*. Wiley.

Frazier, P.I., W. B. Powell, S. Dayanik. 2008. A knowledge gradient policy for sequential information collection. *SIAM Journal on Control and Optimization* **47**(5) 2410–2439.

Frazier, P.I., Jing Xie, S.E. Chick. 2011. Value of information methods for pairwise sampling with correlations. S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, M. Fu, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 3979–3991.

Fu, M.C. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* **14**(3) 192–215.

Fu, M.C., J.Q. Hu, C.H. Chen, X. Xiong. 2004. Optimal computing budget allocation under correlated sampling. R.G. Ingalls, M.D. Rossetti, J.S. Smith, B.A. Peters, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 595–603.

Gelman, A.B., J.B. Carlin, H.S. Stern, D.B. Rubin. 2004. *Bayesian data analysis*. 2nd ed. CRC Press, Boca Raton, FL.

Gupta, S.S., K.J. Miescke. 1996. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference* **54**(2) 229–244.

Hong, L.J., B.L. Nelson. 2006. Discrete optimization via simulation using COMPASS. *Operations Research* **54**(1) 115–129.

Hong, L.J., B.L. Nelson, S.G. Henderson, J. Xie. 2012. `http://simopt.org/wiki/index.php?title=Assemble_to_order`, Accessed 1 July 2013.

Hu, J., Y. Wang, E. Zhou, M.C. Fu, S.I. Marcus. 2012. A Survey of Some Model-Based Methods for Global Optimization. *Optimization, Control, and Applications of Stochatic Systems: in honor of Onésimo Hernández-Lerma*. Birkhäuser, 157–179.

Huang, D., T.T. Allen, W.I. Notz, N. Zeng. 2006. Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. *Journal of Global Optimization* **34**(3) 441–466.

Jones, D.R., M. Schonlau, W.J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* **13**(4) 455–492.

Kim, S.H. 2005. Comparison with a standard via fully sequential procedures. *ACM TOMACS* **15**(2) 155–174.

Kim, S.H., B.L. Nelson. 2006. Selecting the best system. S.G. Henderson, B.L. Nelson, eds., *Handbook in Operations Research and Management Science: Simulation*. North-Holland, 501–534.

Nakayama, M.K. 2000. Multiple comparisons with the best using common random numbers for steady-state simulations. *Journal of Statistical Planning and Inference* **85**(1-2) 37–48.

Nelson, B.L., F.J. Matejcik. 1995. Using Common Random Numbers for Indifference-Zone Selection and Multiple Comparisons in Simulation. *Management Science* **41**(12) 1935–1945.

Oksendal, B. 2003. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Berlin.

Raiffa, H., R. Schlaifer. 1961. *Applied Statistical Decision Theory*. Harvard University.

Rasmussen, C.E., C.K.I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Schruben, L.W., B.H. Margolin. 1978. Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. *JASA* **73**(363) 504–525.

Scott, W., P.I. Frazier, W.B. Powell. 2011. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization* **21** 996–1026.

Shi, L., S. Ólafsson. 2000. Nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability* **2**(3) 271–291.

Tew, J., J.R. Wilson. 1992. Validation of simulation analysis methods for the Schruben-Margolin correlation-induction strategy. *Operations Research* **40**(1) 87–103.

van Beers, W.C.M., J.P.C. Kleijnen. 2008. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *EJOR* **186**(3) 1099–1113.

Villemonteix, J., E. Vazquez, E. Walter. 2009. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* **44**(4) 509–534.

Wang, H., R. Pasupathy, B.W. Schmeiser. 2013. Integer-ordered simulation optimization using R-SPLINE: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM TOMACS* **23**(3) in press.

Wang, Y., M.C. Fu, S.I. Marcus. 2010. Model-based evolutionary optimization. B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, E. Y ucesan, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 1199–1210.

Xu, J., B.L. Nelson, L.J. Hong. 2010. Industrial Strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM TOMACS* **20**(1) 3.

Yang, W.N., B.L. Nelson. 1991. Using Common Random Numbers and Control Variates in Multiple-Comparison Procedures. *Operations Research* **39**(4) 583–591.

Zhou, E., M.C. Fu, S.I. Marcus. 2008. A particle filtering framework for randomized optimization algorithms. S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, J.W. Fowler, eds., *Proc. Winter Simulation Conference*. IEEE, Piscataway, NJ, 647–654.